CODING AND EQUALISATION FOR FIXED-ACCESS WIRELESS SYSTEMS

A Thesis presented for the degree of
Doctor of Philosophy
in
Electrical and Electronic Engineering
at the
University of Canterbury
Christchurch, New Zealand

by

Katharine Ormond Holdsworth,
B.E.(Hons), M.E.

January 2000

For my Parents

ABSTRACT

This Thesis considers the design of block coded signalling formats employing spectrally efficient modulation schemes. They are intended for high-integrity, fixed-access, wireless systems on line-of-sight microwave radio channels. Multi-dimensional multilevel block coded modulations employing quadrature amplitude modulation are considered. An approximation to their error performance is described and compared to simulation results. This approximation is shown to be a very good estimate at moderate to high signal-to-noise ratio. The effect of parallel transitions is considered and the trade-off between distance and the error coefficient is explored. The advantages of soft- or hard-decision decoding of each component code is discussed.

A simple approach to combined decoding and equalisation of multilevel block coded modulation is also developed. This approach is shown to have better performance than conventional independent equalisation and decoding. The proposed structure uses a simple iterative scheme to decode and equalise multilevel block coded modulations based on decision feedback. System performance is evaluated via computer simulation. It is shown that the combined decoding and equalisation scheme gives a performance gain of up to 1 dB at a bit error rate of 10^{-4} over conventional, concatenated equalisation and decoding.

CONTENTS

ACKNOWLEDGEMENTS

I wish to thank the many people who have supported and encouraged me throughout my time as a Ph.D. student and for their contribution to my research – both directly and indirectly. It has been my privilege to take part in this doctoral program.

I am indebted to, and would especially like to acknowledge, Professor Desmond P. Taylor, my supervisor at the University of Canterbury, who facilitated and guided my research. I very much appreciate his time, effort and unfailing enthusiasm, as well as his attentive proof-reading of this Thesis and the interim material. Professor Taylor also provided the funding to attend the following conferences – for which I am grateful: Globecom'98, Sydney, Australia; ISWC'99, Victoria, Canada; and ICC'99, Vancouver, Canada.

My utmost thanks must also be given to my co-supervisor, Dr. Royce T. Pullman, Vice President Research and Development, Digital Microwave Corporation (DMC), Wellington – formally MAS Technology Ltd – for his guidance and careful editing of successive drafts of research papers and this Thesis.

I am also grateful to Neville Jordan, who originally seeded the idea of undertaking a Ph.D. in conjunction with MAS Technology Ltd, for his ongoing interest.

I wish to acknowledge and thank both the Foundation for Research, Science and Technology for providing me with a Graduates in Industry Fellowship (GRIF) as well as DMC for their financial support. I am also grateful for the time I have spent based at DMC – as part of the GRIF scheme – for the resources they provided and the interest and encouragement from my colleagues.

My thanks must also go to two visitors to the Electrical and Electronic Engineering Department at the University of Canterbury: Professor Bill H. Tranter, visiting Erskine Fellow, for his initial guidance, encouragement and interesting technical discussions at the beginning of my work; and Dr. Wolfgang Gerstacker, visiting Post-Doctoral Fellow in the last few weeks of my research, whose infectious enthusiasm for this topic and technical discussions were invaluable.

I would also like to thank the post-graduates and staff at the University of Canterbury with whom I have been associated, for helping to provide the stimulating environment in which to work.

Finally, I would like to thank my family and friends – especially my wonderful parents who also proof-read my Thesis – for their ongoing support, love and encouragement.

Katharine O. Holdsworth

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF ABBREVIATIONS

A/D	analogue-to-digital
ARQ	automatic repeat request
AWGN	additive white Gaussian noise
BCH codes	Bose-Chaudhuri-Hocquenghem codes
BER	bit error rate
CDMA	code-division multiple access
dB	decibels
DFE	decision feedback equalisation
E-Commerce	Electronic Commerce
FEC code	forward error correcting code
FIR	finite impulse response
ISI	intersymbol interference
LMS	least-mean-square
LSB	least significant bit
M-ary QAM	a QAM constellation containing M signal points
MAP	maximum *a posteriori* probability
MLSE	maximum likelihood sequence estimation
MMSE	minimum mean-squared error
MSB	most significant bit
MSED	minimum squared Euclidean distance
PAM	pulse amplitude modulation
PAR	peak-to-average power ratio

PSD	power spectral density
PSK	phase-shift keying
QAM	quadrature amplitude modulation
QoS	quality of service
QPSK	quaternary-PSK
RSSE	reduced-state sequence estimation
SNR	signal-to-noise ratio
TCM	trellis-coded modulation
TDMA	time-division multiple access

GLOSSARY OF MATHEMATICAL TERMS

$A_{d_{min}}$	error coefficient of a code
$A_{d_{min},\nu}$	error coefficient for code C_ν
A_ω	weight distribution of a code
$\bar{\mathbf{A}}_j$	output binary data block from the j^{th} iteration
a_i and b_i	pair of independent integers relating to desired signal point
$\mathbf{a}$	vector of uncoded bits
b_ν	number of encoded bits per two-dimensional symbol
C	block code; $C = (n, k, d_{min})$
C_ν	the ν^{th} component code representation; $C_\nu = (n_\nu, k_\nu, d_{min,\nu})$
$C(f)$	equivalent baseband channel frequency response
$\mathbf{C}$	hierarchy of component codes in a multilevel code
$c(t)$	channel impulse response
c_κ	amplitude of channel path
$\mathbf{c}$	encoded code vector
D_ν	decoder for C_ν
$D^2_{min}(C_\nu)$	MSED of the coded partition level $S_{\nu-1}/S_\nu$
$D^2_{min}(\mathbf{C})$	MSED of a multilevel code
d^2_{ref}	MSED between signal points in an uncoded reference constellation
d_{min}	minimum Hamming distance
$d_{min,\nu}$	minimum Hamming distance of component code, C_ν
$E\{\bullet\}$	expected value
E_b	average received bit energy

E_{min}	energy of symbol with lowest amplitude
E_s	received signal energy
E_{sym}	received multidimensional symbol energy
e_i	residual error
$e_{i,j}$	residual error during the j^{th} iteration
$\mathbf{e}$	error pattern vector
$F(f)$	frequency response of the pulse shaping filter
$\bar{F}(f)$	overall pulse shaping filter response
$F_{RC}(f)$	frequency response of a raised cosine pulse shape
$F_{SRRC}(f)$	frequency response of a square-root raised cosine pulse shape
$\mathcal{F}\{x\}$	Fourier transform of the function x
f_c	carrier frequency
$f_{Nyquist}(t)$	impulse response of a sinc pulse
$f_{RC}(t)$	impulse response of a raised cosine pulse shape
$f_{SRRC}(t)$	impulse response of a square-root raised cosine pulse shape
$f(t)$	transmitter impulse response
$f^*(-t)$	impulse response of the receiver matched filter
$\mathbf{G}$	code generator matrix
$\mathbf{G}_j$	j^th code generator vector
$H(f)$	overall channel frequency response
$\mathbf{H}$	parity check matrix
$h(t)$	overall channel impulse response
i	discrete time index
k	number of original uncoded data bits transmitted
k_ν	number of original uncoded data bits in code C_ν
M	number of signal points in constituent constellation
m	number of set partition levels
$m(t)$	complex baseband signal
N	number of two-dimensional constellations in a multidimensional constellation

N_n	average number of nearest neighbours of each signal point
N_0	power spectral density of the noise
N_B	number of taps in the feedback filter
N_F	number of taps in the forward filter
N_κ	total number of channel paths
n	total number of bits in coded data sequence
n_ν	total number of code bits in code C_ν
$n(t)$	Gaussian noise component of matched filtered received signal
n_c	common block length of component codes
n_d	size of transmitted binary data block
$n_I(t)$	in-phase noise component
$n_Q(t)$	quadrature noise component
P_e	error performance of a multilevel code
$P_e(C_\nu)$	error performance of component code C_ν
P_s	probability of a symbol error
$P(f)$	frequency response of DFE feedback filter
$P(x_i \vert z_{i,1})$	the *a posteriori* probability of x_i given $z_{i,1}$
$\mathbf{P}_i$	complex feedback tap weights at the i^{th} sampling interval
$\mathbf{P}_{i,j}$	complex feedback tap weights at the i^{th} sampling interval during the j^{th} iteration
$p\big(n(t)\big)$	complex Gaussian probability function
Q	number of bits assigned to each symbol
$Q\,(\bullet)$	Gaussian error probability
R	code rate (ratio of number of data bits to total bits in code word)
$R_{\mathbf{C}}$	effective code rate of a multilevel code (bits/symbol)
R_0	actual data rate (bits/symbol) at output of channel encoder
R_s	channel or data rate (bits/symbol) of modulation scheme
R_ν	rate of component code C_ν
$\mathbf{R}_i$	samples at time i contained in the forward filter

$\mathcal{R}(r,m)$	Reed-Muller code designation – r is the code order and 2^m is the code block length
r	code order
$r(t)$	received signal
$\mathbf{r}$	received code vector
SNR	SNR calculated at matched filter output in terms of $\frac{E_b}{N_0}$
SNR_0	SNR calculated at matched filter output in terms of $\frac{E_s}{N_0}$
S_0	original two-dimensional constituent signal constellation
S_l	partitioned signal subset
$S_{\nu-1}/S_\nu$	the ν^{th} constellation partition
$\mathcal{S}_w(f)$	power spectral density of $w(t)$
s_i	current output of forward filter
$s_{i,j}$	current output of forward filter during the j^{th} iteration
$\mathbf{s}$	syndrome vector of $\mathbf{r}$
T	symbol period
T_b	bit period
t	time
$\mathbf{W}_i$	complex forward tap weights at the i^{th} sampling interval
$W(f)$	frequency response of DFE forward filter
$w(t)$	AWGN component
$\hat{\mathbf{X}}_i$	samples at time i contained in the feedback filter
$\hat{\mathbf{X}}_{i,1}$	samples at time i contained in the feedback filter in first iteration
$\bar{\mathbf{X}}_{i,j}$	samples contained in the feedback filter at time i during the j^{th} iteration; here, $j \neq 1$
$\bar{\mathbf{X}}_{j+1}^{tot}$	the re-encoded symbols from the j^{th} iteration
x	ratio of average energy per block to the MSED of the code
x_i	(complex) encoded modulated symbols
$\hat{x}_i$	hard symbol decision on z_i
$\hat{x}_{i,1}$	hard symbol decision on $z_{i,j}$ during the 1^{st} iteration
$\bar{x}_{i,j}$	re-encoded symbol values during the j^{th} iteration; here, $j \neq 1$

$\tilde{x}_{i,j}$	MMSE estimator of x_i during the j^{th} iteration
Z^2	square integer lattice
Z^{2N}	integer lattice
z_i	estimated, equalised version of transmitted symbol x_i
$z_{i,j}$	estimated, equalised symbol during the j^{th} iteration
$\otimes$	convolution operator
$\prod$	product operator
$\sum$	summation operator
$*$	complex conjugate operator
X	Cartesian product operator
β	roll-off parameter, or excess bandwidth, of transmit pulse shapes
γ_0	nominal coding gain
δ	minimum Euclidean distance
δ_l^2	MSED between elements in a partitioned signal subset, S_l
δ_0^2	MSED between elements in S_0
η	spectral efficiency
μ	step size parameter to ensure LMS algorithm stability
π	Pi, 3.14159
ν	component code index
ρ	throughput (bits/symbol)
σ_n^2	noise variance
τ_κ	relative delay of channel path number κ
ω	(Hamming) weight of a code word

Chapter 1

INTRODUCTION

But let your communication be, Yea, Yea; Nay, Nay:
for whatsoever is more than these cometh of evil.
– *The Gospel According to St. Matthew (5:37)*

It is dangerous to put limits on wireless.
Guglielmo Marconi (1932)

1.1 General Introduction

Communication technology is advancing at a phenomenal rate and has become an extremely exciting and important field of research and development. Originally communication was limited to a (now) unacceptable quality of voice over an analogue telephony or radio network. With the advent of various forms of digital communication and its overwhelming technological and economic advantages, digital is superseding analogue. Digital networks now successfully transmit high-integrity digital information in the form of computer generated data and other forms of digitised data – such as audio and video.

The progression in digital communication technology has enabled the general population to have access to a range of useful and valuable communication tools at an affordable cost. There has been a huge growth in popularity of the Internet, personal communicators and, more recently, E-Commerce [1]. These are obvious examples of key market forces driving the intrinsic technological need for significant amounts of data to be transmitted quickly and efficiently. This is the galvaniser for newly industrialising countries – which previously had low levels of reliable communication – to install modern communication facilities at an ever increasing rate so that they can both operate and compete in the globalised industrial economy. Furthermore, in many countries, the communications market has been (or is being) liberalised. Telecommunication companies, which were formerly government-owned or state-regulated, are now being privatised [2]. The new competitive communications industry has allowed both wire-line and wireless communications companies to compete with existing wire-line networks – where the latter networks were previously "natural monopolies" [2].

Since competition has opened the industry to market forces, it has resulted in heavy investment into network, equipment capabilities and quality of service (QoS) as companies and countries strive to upgrade existing systems in order that they remain or become competitive. Many telecommunication companies have also digitised and expanded their networks to keep abreast of customer QoS demands and changing technology [3]. Equipment manufacturers and suppliers are also striving for extremely competitive products in terms of cost and overall performance.

The original use of line-of-sight fixed-access, wireless microwave radio was for telephony and began in the early 1950's using analogue modulation schemes [4]. Since the early 1970's these systems have also been utilised for digital communication [3]. The pioneering digital systems used modest spectral efficiencies and typically operated over short distances [3, 4]. Historically, these networks provided an extension for wire-line telephony networks [5]. As wireless technology has improved, capacity has increased [3]. The design of a wireless network is fast – it can be installed and operating in a fraction of the time and price it takes to install a (copper or fibre) wire-line network [6, 7]. A basic advantage of wireless networks is that they may be designed for current demand, then (given the available spectrum) quickly and cheaply scaled up to incorporate expanding capacity demands. Wire-line networks, on the other hand, typically need to be over engineered to meet future capacity requirements, and therefore, are time consuming and expensive to install [6].

Over the past ten years, wireless networks have been gaining momentum and acceptance as a viable (and often the only) option for some urban and rural applications in both developed and developing countries [7]. Each network opportunity can serve a different purpose – for example, in developed countries wireless networks may provide last-mile access to homes and businesses for broadband applications [7]. Because of relatively low cost and ease of operation, they are becoming the preferred option for many customers. In rural and isolated areas where wire-line is expensive or physically difficult to install, wireless provides a viable solution for basic communication requirements. In developing countries where copper wire-line infrastructure is often

non-existent, wireless networks are cost-effective solutions to assist in bringing basic telephony networks into the 21^{st} century [6–8]. For example, 75% of the population of India live in villages and many have no, or extremely limited, telecommunications facilities [9]. Employing fixed-access wireless, it is possible to quickly deploy basic services to this sector of the population using limited financial resources. In each case, wireless access is recognised as providing reliable, efficient and cost effective communication.

Wireless data transmission is becoming the new international trend in progressing towards a "knowledge economy" and advancing the E-Commerce industry. With the advent of new digital techniques, the capacities and capabilities of wireless networks have greatly increased the range of services offered. Fixed-access wireless systems operating alone or in conjunction with other systems will play a vital role in this trend. In the case of mobile networks, wireless communications is a prerequisite.

In advanced global digital communication scenarios, there are increasing demands from present users and forecasts for a large number of potential future users. These demands necessitate higher data rates on the existing and increasingly crowded lower frequency bands. Consequently, high-capacity broadband wireless communication has a strong focus. The recent ITU global standard, International Mobile Telecommunications-2000 (IMT-2000) [10] defines air interface techniques for future high-rate wireless systems. This standard aims to provide unconstrained and seamless public access to all forms of communication via wireless media. To ensure that wireless providers are able to offer high-integrity performance levels in the already limited spectrum and

inhospitable wireless environment, existing systems will require advances in signal processing, modulation, coding and radio-frequency technologies.

In the design of any digital system there are system trade-offs required to reach the desired performance levels. In general, there are four major design goals [11]:

1. to minimise the probability of received bit error;
2. to minimise the transmitted power;
3. to maximise the bandwidth efficiency;
4. to minimise the equipment complexity.

These system constraints are conflicting. However, they may be addressed by using sophisticated coding techniques and large bandwidth efficient signalling constellations. There are also other system goals, such as minimising receiver decoding delay and system cost.

Ideally, a communication system should transmit error-free data from one point to another. However, as this data is transmitted over the channel, whether it be wired or wireless, it may be exposed to elements which degrade its quality to an extent it may no longer be understood by its receiver. Channel coding is introduced into a system to detect or to alleviate erroneous channel behaviour. In 1948 Shannon [12] developed a noisy channel theorem stating that an arbitrarily small probability of error can be achieved at rates below channel capacity, C. The channel capacity, C, of a channel perturbed by additive white Gaussian noise (AWGN) and limited to a bandwidth of B-Hz is given by [13]

$$C = B \log_2 \left(1 + \frac{S}{N}\right) \text{ bits/second} \tag{1.1}$$

where S is the signal power; N is the noise power of the system; and $\frac{S}{N}$ is the signal-to-noise power ratio at the receiver. Equation (1.1) is often referred to as the Shannon-Hartley law [14]. Although this theorem provides an upper bound on transmission rate and indicates the existence of codes capable of reaching this capacity with an arbitrarily small error probability, it does not indicate how this limit is to be achieved [15].

Shannon's theorem, combined with the obvious commercial drive to reach full achievable capacity over a given channel bandwidth, established a challenge for researchers. As was recently so succinctly recorded in [16],

> *"Claude Shannon in 1948 had proven the existence of error-correcting codes that, under suitable conditions and at rates less than channel capacity, would transmit error-free information for all practical applications. The hunt for optimal, mechanisable error-correcting codes was on!"*

Soon after Shannon's declaration, the first practical error correcting codes were suggested by Hamming [17] and Golay [18]. Interestingly, the first error correcting code was not developed with communications in mind. It was developed as a result of the frustration Hamming felt when the mechanical relay model computer he used at the Bell Telephone Laboratories crashed on detecting an error [17, 19]. To compound the matter, these computers were unattended during evenings and weekends and would often come to a halt, thereby losing valuable working time [17]. As Hamming recalled years later in an interview [19]:

"Two weekends in a row I came in and found all my stuff had been dumped and nothing was done. I was really aroused and annoyed because I wanted those answers and two weekends had been lost. And so I said 'Damn it, if the machine can detect an error, why can't it locate the position of the error and correct it?'."

The first class of error correcting codes – single error correcting block codes [17] – were born from this problem. Over the past 50 years error control coding has continued to develop following the work of Shannon, Hamming and Golay. Some of the most notable coding theory milestones since Hamming were [15, 20]: in 1954, the work of Reed [21] and Muller [22] resulted in what are now known as Reed-Muller codes. Reed-Muller codes were a new and important step in coding history as they had a great flexibility in correcting a varying number of errors [23]. In 1955, convolutional codes – an alternative to block codes – was published by Elias [24]. In 1959, Bose-Chaudhuri-Hocquenghem (BCH) codes, a class of multiple error correcting block codes, were independently researched by Bose/Chaudhuri and Hocquenghem [20]. Reed and Solomon developed an extremely powerful block coding scheme well suited to burst error correction in 1960 [16]; in 1966, Forney discussed the use of concatenated coding [25]; and in 1967, Viterbi suggested the trellis decoding algorithm which provided a breakthrough in decoding techniques [26]. Imai and Hirakawa presented in 1977 a multilevel coding scheme whereby the signal set is partitioned and one code is associated with each partition [27]; and in 1982, Ungerboeck published a paper in which he demonstrated how to efficiently combine coding and modulation using set partitioning in a scheme

known as trellis-coded modulation (TCM) [28]. There have been many other important historical coding developments, however, the papers mentioned here have been of real and lasting consequence to modern coding theory. The latest major coding development has been fuelled by the introduction, in 1993, of Turbo codes and iterative decoding [29]. Each coding milestone has resulted in performance approaching closer to the original bound developed by Shannon, with Turbo codes being within less than one decibel at an error rate of 10^{-5}. Although the origins of error control coding were an engineering problem, the subject has since been developed through the use of advanced mathematical techniques [30] combined with computer simulations and code search algorithms.

The goal of this Thesis is to investigate high-capacity, bandwidth efficient coded signalling structures using relatively simple encoding and decoding techniques. Since there is a significant interest in fixed-access wireless systems via frequency-selective channels, we also investigate an approach to combined equalisation and decoding to improve system performance over conventional schemes. Combined equalisation and decoding may also be applied in contexts other than fixed-access wireless. It is one method of improving overall system performance without necessarily increasing the complexity of the coding structure.

1.2 System Goals

We specifically consider the design of coded signalling formats for high-integrity, fixed-access wireless systems on line-of-sight microwave radio chan-

nels. The systems of interest usually have strict bandwidth and power limitations, use low gain antennas and often require significant coding gain with a short decoding delay to achieve the necessary performance levels. The tightly constrained available bandwidth and the requirement to maximise system capacity indicate the desirability of minimising any spectral expansion due to coding.

In some digital communication scenarios, protocols transmit data in fixed length blocks for error correcting purposes. In these cases, block codes are more effective than convolutional codes as they do not require truncation between blocks using tailing symbols – which are not bandwidth efficient. The coding signalling structures we consider are high-rate, multilevel block coded modulation formats [31, 32]. They may be constructed using an arbitrary hierarchy of codes mapped to an expanded signal set and offer significant coding gain with modest decoding complexity [31,32]. Multilevel codes provide an alternative to conventional TCM [28] in the construction of bandwidth efficient coded signal sets.

Multilevel codes have been studied widely [27,31–33]. The present work investigates a specific family of multilevel block coded modulation based on a hierarchy of Reed-Muller component codes. It also investigates an analytical approximation to the error performance of multilevel block coded modulation employing quadrature amplitude modulation (QAM) and a hierarchy of Reed-Muller block codes [34, 35]. This analytical approximation is compared to simulation results and is shown to be a good performance estimate at moderate to high signal-to-noise ratio (SNR). Component codes in a multilevel code

construction may be soft- or hard-decision decoded. We investigate the relative advantages of soft- or hard-decision decoding each component code in terms of performance, delay and system complexity.

The fixed-access wireless channel introduces phase and amplitude distortions to the transmitted signal that result in intersymbol interference (ISI) [36]. In practice, we do not have prior knowledge of the channel characteristics and thus we employ adaptive equalisation [37] – in particular, decision-feedback equalisation (DFE) – to mitigate channel effects [13, 37, 38].

Traditionally, channel equalisation has been implemented separately from decoding. Following the Bayesian DFE concepts of [39, 40] and the joint equalisation and decoding concepts of [40, 41], we develop a simple iterative approach to combined equalisation and multilevel block decoding. Essentially, we combine the DFE [37] and decoding of multilevel block codes mapped to large QAM constellations [31, 33, 34] through the use of an iterative technique [42]. Following [43], after the training sequence, it is clear that the forward and feedback filters of the DFE may be (adaptively) optimised independently. Assuming the optimal forward filter, we may then combine the feedback filter and the decoder in an independent iterative structure. The decoder output for each block is re-encoded, remodulated, processed repeatedly through the feedback filter of the DFE and then re-decoded. As will be seen, there is a small but definite performance improvement on each iteration. The proposed structure achieves a performance gain of up to 1 dB at a bit error rate (BER) of 10^{-4} over conventional, independent equalisation concatenated with soft-decision decoding.

1.3 Thesis Contents

Chapter 2 provides some general introductory background information on the communications system. This Chapter develops the complex baseband communications model and system notation used throughout the Thesis. The operation of the transmitter, the fixed-access line-of-sight wireless channel model and common noise sources are discussed. We review QAM constellation construction and show how Ungerboeck's set partitioning [28] and multidimensional shaping codes [44] may be applied to these constellations. Chapter 2 also discusses error correcting coding principles and considers the construction of linear block codes. We particularly focus on Reed-Muller block codes [23,45] which are employed in the examples considered later in the Thesis. Finally, this Chapter provides a literature review outlining previous work related to multilevel coding and combined equalisation and decoding.

Following previous work in this area [27,31,44,46], Chapter 3 describes bandwidth efficient multilevel coded signalling formats based on signal set partitioning. Large QAM constellations are employed and shaped using a multidimensional shaping code and a hierarchy of component codes. The staged structures used to encode and decode multilevel block codes are also discussed.

Chapter 4 evaluates the performance of various multilevel block codes employing several QAM constellations and various Reed-Muller component code hierarchies. The channel of interest here is the AWGN channel. An analytical approximation to the error performance of these multilevel block codes is described and compared to simulation results. From the analytical perspective, we consider the effect of parallel transitions. The trade-off between the

overall code distance and the error coefficient of the code is also explored.

Chapter 5 describes a simple and novel approach to combined decoding and equalisation of multilevel coded modulation. We show that this approach has better performance than conventional independent equalisation and decoding. The proposed structure uses a simple iterative scheme to decode and equalise multilevel block coded modulations based on decision feedback. System performance is evaluated via computer simulation and discussed in Chapter 6. In this Chapter, we show the performance gains when using the iterative combined equalisation and decoding scheme as opposed to the conventional, concatenated system. As noted, we use several Reed-Muller component code hierarchies as examples and consider various large QAM constellations.

The final Chapter concludes the achievements and contributions of this Thesis and outlines some topics of interest relating to this area for future research.

1.4 Thesis Contributions

The original contributions of this Thesis concentrate on two major areas: 1) multilevel block code design and performance approximation; and 2) combined equalisation and multilevel block decoding. Here we outline the contributions which are considered to be original to this Thesis. The publications from this research and submissions currently under consideration for publication are also cited.

1.4.1 Contributions to Multilevel Coding

In the multilevel coding subject area, the original contributions revolve around the discussion of specific multilevel block codes and their error performance. We employ large multidimensional QAM constellations and as examples, use hierarchies of Reed-Muller component codes. The analytical approximation to the error performance of these codes is described and compared to simulation results. In two short papers published in this area we have shown that the analytical error approximation is a good estimate, at moderate to high SNR, of simulated performance. The effect on error performance of parallel transitions and the trade-off between the minimum squared Euclidean distance of the overall code and its error coefficient are considered. This research is reported in detail in Chapter 4 and summarised in the following published papers.

- K.O. HOLDSWORTH, D.P. TAYLOR AND R.T. PULLMAN, "On the Error Performance of Multilevel Block-Coded Modulation", *IEEE Communications Letters*, vol. 3, no. 11, pp. 311-313, November 1999 [34].

- K.O. HOLDSWORTH, D.P. TAYLOR AND R.T. PULLMAN, "On the Error Performance of Multilevel Block Coded Modulation", *ISWC'99 – IEEE International Symposium on Wireless Communications*, pp. 21-22, June 1999 [35].

1.4.2 Contributions to Combined Equalisation and Decoding

The original contribution in the combined equalisation and decoding realm is to develop an approach to iterative combined equalisation and decoding of multilevel block coded modulation. We show that this approach has better performance than conventional, independent equalisation and decoding. The proposed structure uses a simple iterative scheme to decode and equalise multilevel block coded modulations based on decision feedback. One of the notable differences of this scheme compared to others in the literature is that it does not require the use of interleaving. System performance of this scheme is evaluated via computer simulation. We show that the combined decoding and equalisation scheme gives a performance gain of up to 1 dB at a bit error rate of 10^{-4} over conventional, concatenated equalisation and decoding. This work is covered in detail in Chapters 5 and 6. It is also summarised in the following papers.

- K.O. HOLDSWORTH, D.P. TAYLOR AND R.T. PULLMAN, "On Combined Equalization and Decoding of Multilevel Coded Modulation", accepted for *ICC'00 – IEEE International Conference on Communications*, session 31.8, June 2000 [47].

- K.O. HOLDSWORTH, D.P. TAYLOR AND R.T. PULLMAN, "On Combined Equalization and Decoding of Multilevel Coded Modulation", submitted to *IEEE Transactions on Communications*, November 1999.

Chapter 2

BACKGROUND INFORMATION

2.1 Introduction

A digital communications system is required to transmit and receive information with high integrity, without substantial delay (especially in the case of voice) and in a cost effective manner. The data source may represent any number of information sources – for example, voice, video or computer generated data.

A block diagram showing signal flow through a typical communications system is shown in Figure 2.1. We have separated the system into three major functional blocks: the transmitter; the channel (including additive noise); and the receiver. This Chapter provides some general background information on the communications system, discussing elements of Figure 2.1. It sets notation and basic system information for the remainder of the Thesis.

As shown in Figure 2.1, the transmitter's initial task is to arrange the sequence of analogue or digital data from the information source into a suit-

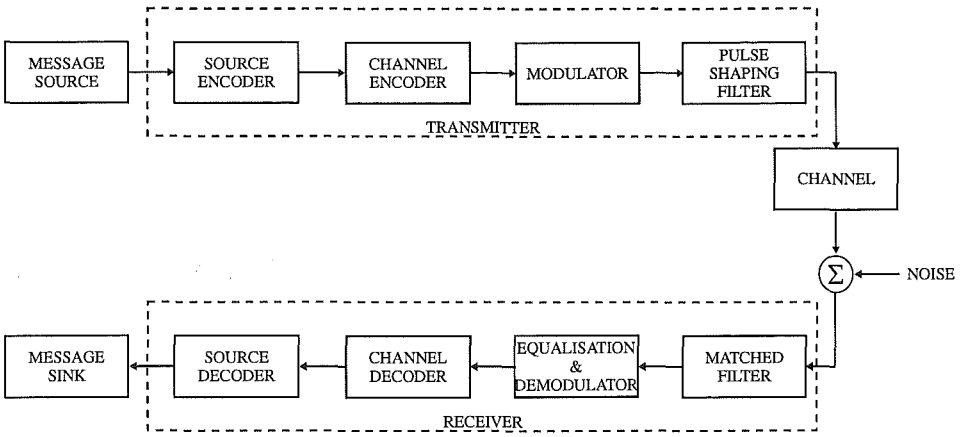


Figure 2.1: Block diagram of a communications system.

able transmission format – for example, analogue-to-digital (A/D) conversion for analogue sources [13]. This function is commonly referred to as source encoding. We assume here that the bit streams of ones and zeros at the output of the source encoder are produced with equal probability regardless of the type of information source. The digitised data is then delivered to the channel encoder. The output of the channel encoder is modulated onto the chosen signal constellation and then pulse shaped for transmission over the channel.

Once the signal has been prepared for transmission by the transmitter, it is passed to the channel. In this Chapter, we focus on channel models most relevant to fixed-access wireless channels, namely the AWGN channel and the multipath fading channel. The AWGN channel is the most well known channel model and is often used to test optimal system performance. The slow fading

multipath channel is a realistic model for fixed-access point-to-point microwave radio links [48].

The receiver then processes the received signal to estimate the transmitted data. The signal is demodulated and passed through a matched filter matched to the transmitter pulse shape. To recover the data from the received signal which has been impaired by ISI due to multipath channel propagation, equalisation is performed. The equaliser used in this Thesis is the decision-feedback equaliser. The equaliser is a common component in a receiver model and its principles are outlined in this Chapter.

We also introduce coded modulation principles. These provide an introduction into some of the concepts used in multilevel coding. Finally, we provide a brief literature review outlining the relevant literature in the multilevel coding and combined equalisation and decoding areas – these are the two areas in which this Thesis makes an original contribution.

Throughout the Thesis, for both analysis and simulation we assume a complex baseband communications model. Baseband models are usually employed when utilising computer simulations to test system performance. In a practical system, the transmitter will translate the modulated signal to its allotted frequency band before transmitting over a channel.

2.2 Quadrature Amplitude Modulation

A commonly used form of modulation in high-capacity radio communication systems is M-ary quadrature amplitude modulation (QAM) [49], which will be assumed throughout this Thesis. QAM is an efficient way to transmit

many bits per symbol. It makes efficient use of the available transmission bandwidth by using a multilevel, two-dimensional signalling scheme. It combines amplitude and phase modulation by employing two carriers, $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$, and has the general form [14, 50, 51]

$$x_i(t) = \sqrt{\frac{2E_{min}}{T}} a_i \cos(2\pi f_c t) + \sqrt{\frac{2E_{min}}{T}} b_i \sin(2\pi f_c t), \quad iT \leq t \leq (i+1)T \tag{2.1}$$

where E_{min} is the energy of the signal with the lowest amplitude for each quadrature component; a_i and b_i represent a pair of independent integers chosen in accordance with the location of the desired signal point; t denotes time; f_c is the carrier frequency; T is the symbol period; and i is the discrete time index, such that $i = (0, 1, 2, \cdots)$.

A QAM constellation usually contains $M = 2^Q$ equally spaced signal points, where Q is an integer and defines the number of bits assigned to each symbol. It is usually a subset of signal points from the square integer lattice Z^2 [14, 50–52]. Figure 2.2 shows the signal space allocation for several QAM constellations. The symbol period of a QAM signal point is denoted $T = QT_b$ – where T_b is the bit period; and Q is the number of bits per symbol.

Each signal point in a QAM constellation may be uniquely labelled using Q bits. There are a number of methods to assign bit labels to each signal point. For example, we may randomly map bits to signal points; use Gray mapping [14]; or use Ungerboeck mapping by set partitioning [28]. The throughput, ρ, of an uncoded QAM system is $\rho = Q$ bits/symbol.

When implementing a QAM constellation, the more signal points that are used, the higher the spectral efficiency, η, of the signal set. However, the

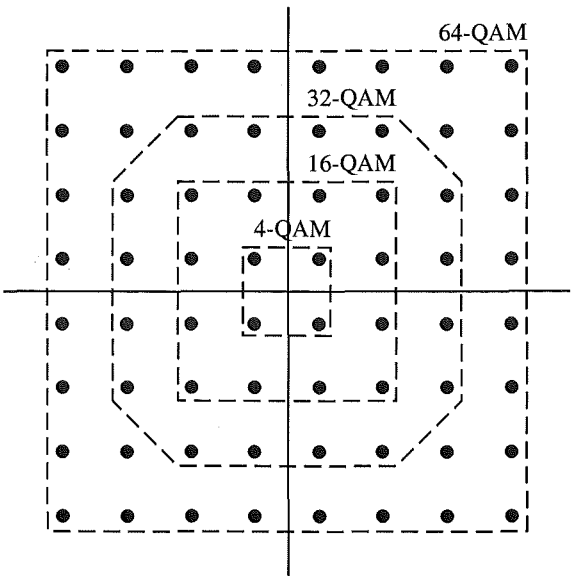


Figure 2.2: Square signal space constellations for several QAM schemes.

average power of each constellation size is constrained to an average signal power of, say, $E_s = 1$. Thus, increasing the constellation size results in the signal points becoming closer together, which increases the probability that receiver noise or other channel interference will result in symbol errors [53]. In order to maintain a predefined level of performance when increasing the size of a QAM constellation we need to increase the transmitter power [13]. In the AWGN channel, the probability of making a symbol error, P_s, using QAM is dominated by the minimum Euclidean distance, δ, between signal points such

that [13, 54]

$$P_s \approx N_n \, Q\left(\frac{\delta}{\sqrt{2N_0}}\right) \tag{2.2}$$

where N_0 is the power spectral density (PSD) of the noise; N_n is the average number of nearest neighbours of each signal point; and $Q(\bullet)$ is the Gaussian error probability,

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2} dy \tag{2.3}$$

2.3 Pulse Shaping

In order to transmit the encoded and modulated signal points successfully over a channel, we pass the encoder output pulses through a transmitter pulse shaping filter to obtain the complex baseband signal

$$m(t) = \sum_i x_i f(t - iT) \tag{2.4}$$

where x_i is a sequence of (complex) encoded and modulated symbols, such that $x_i = a_i + jb_i$; $f(t)$ is the transmitter pulse shape; and T is the symbol period. At the receiver there is a matched filter matched to the transmitter pulse shaping filter giving an overall pulse shaping filter response,

$$\bar{F}(f) = F(f)F^*(f) \tag{2.5}$$

where $F(f)$ is the frequency response of the pulse shaping filter; and $F^*(f)$ is the complex conjugate of $F(f)$. Here, we assume that

$$\int_{-\infty}^{\infty} |F(f)|^2 df = \frac{1}{T} \tag{2.6}$$

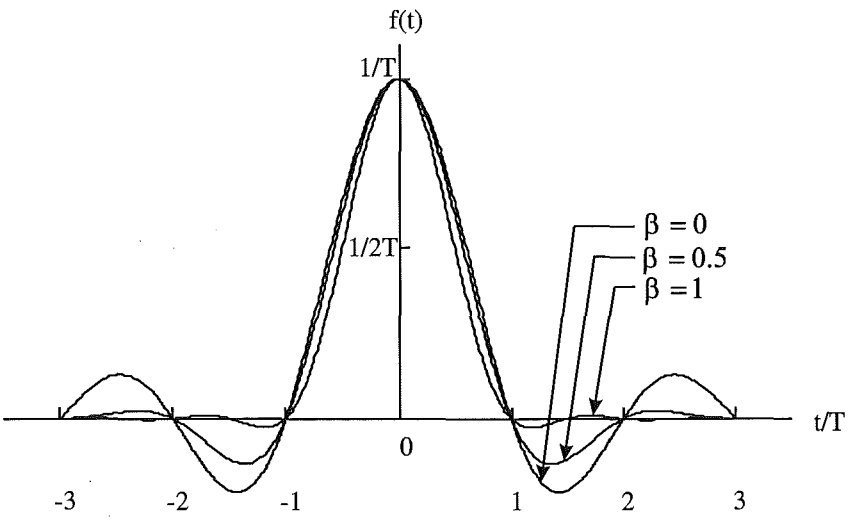
Assuming the receiver has correct sample timing, the overall pulse shaping filter, $\bar{F}(f)$, should minimise bandwidth and exhibit no ISI at the receiver sampling points. That is, the pulse shaping filter should satisfy the first Nyquist criterion for ISI free transmission. The ideal pulse shape complying with the first Nyquist criterion is the sinc pulse which has the impulse response [50]

$$\begin{aligned} f_{Nyquist}(t) &= \frac{1}{T}\frac{\sin(\pi t/T)}{\pi t/T} \\ &= \frac{1}{T}\text{sinc}(t/T) \end{aligned} \tag{2.7}$$

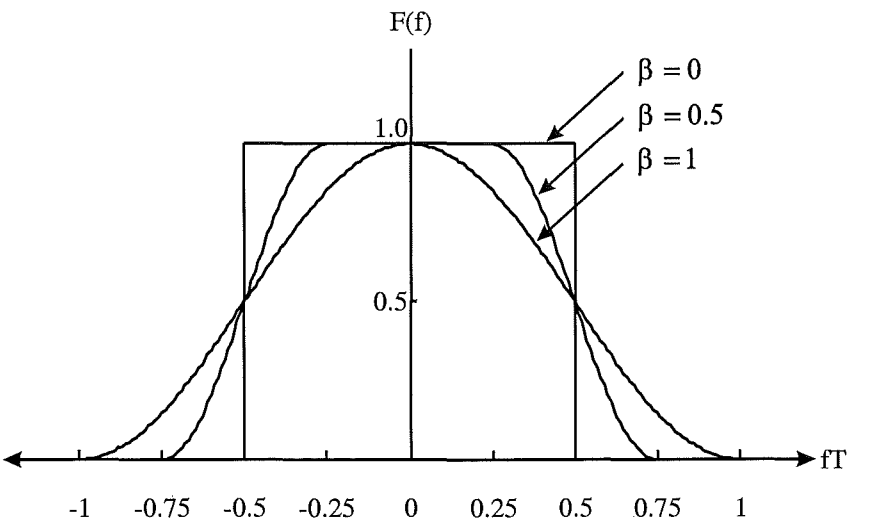
The sinc pulse shape is non-causal [55,56]. To avoid ISI it needs to be sampled once at the peak of the main lobe to get one good sampling point and thereafter at the zero crossings in the tail of the pulse. There are two basic problems associated with the sinc pulse: 1) its frequency spectrum is an ideal rectangular characteristic, which is unrealisable; and 2) its tails in the time domain decay slowly at a rate of only $1/|t|$. Since its rate of decay is slow, there is no margin for error in the receiver sampling times if ISI free reception is to be obtained. Any timing jitter in the sampling may result in very large ISI. Therefore, other more practical pulse shapes are generally employed.

A commonly used, more easily approximated class of unrealisable overall receiver pulse shapes is the raised cosine pulse shape. It also complies with the first Nyquist criterion for ISI free transmission. As shown in Figure 2.3(a) it has the impulse response [14,50],

$$f_{RC}(t) = \frac{1}{T}\frac{\sin(\pi t/T)}{\pi t/T}\frac{\cos(\beta\pi t/T)}{1-4\beta^2t^2/T^2} \tag{2.8}$$



(a) Impulse response (truncated)



(b) Frequency response

Figure 2.3: Raised cosine filter characteristics for different roll-off factors.

and a corresponding frequency response (cf. Figure 2.3(b)) of

$$F_{RC}(f) = \begin{cases} 1 & 0 \leq |f| \leq \frac{(1-\beta)}{2T} \\ \frac{1}{2}\left(1 - \sin\left[\frac{\pi T}{\beta}\left(|f| - \frac{1}{2T}\right)\right]\right) & \frac{(1-\beta)}{2T} < |f| \leq \frac{(1+\beta)}{2T} \\ 0 & |f| > \frac{(1+\beta)}{2T} \end{cases} \tag{2.9}$$

where β is the roll-off parameter or excess bandwidth, $0 \leq \beta \leq 1$. When $\beta = 0$, Equation (2.8) is the Nyquist minimum bandwidth sinc pulse of Equation (2.7). When $\beta = 1$, the pulse shape requires 100% excess bandwidth. The raised cosine pulse shape has zero-crossings at each integer multiple of T (symbol period) as does the sinc pulse. But compared to the sinc pulse, it has excess bandwidth – that is, we trade bandwidth for robustness to timing error. Provided that the receiver samples the signal every T seconds, once at the peak of the main lobe and thereafter at the zero crossings, there will be no ISI caused by the pulse shaping filters. In addition, as $\beta \to 1$ the tails of the raised cosine pulse shape decay at a considerably faster rate than the sinc pulse. Therefore, timing jitter at the receiver does not have such a detrimental effect on system performance. As $\beta \to 0$, the pulse shape tends towards a sinc pulse, therefore any timing jitter will cause increasing ISI[1].

In a communications system we face the problem that the received signal is corrupted by AWGN. By using a receive filter matched to the transmitter pulse shape we may optimise the detection of the data symbols – since the matched filter maximises SNR at the sampling points and helps to eliminate phase distortion [50]. The impulse response of the optimum matched filter is

[1]When $\beta = 1$, the second Nyquist criterion is satisfied, whereby the pulse shape has maximum insensitivity to sampling errors.

a time-reversed, complex-conjugated and delayed version of the transmitted pulse shape. Ideally, we want the combined response of the whole system to achieve zero ISI performance. Here we specify the overall response of the system (transmitter and receiver pulse shapes), $\bar{F}(f)$, to be a raised cosine pulse shape. To achieve this, we design the transmitter pulse shaping filter and the receiver matched filter to be square-root raised cosine pulse shapes. The impulse response of the square-root raised cosine pulse shape may be expressed as [57],

$$f_{SRRC}(t) = \frac{1-\beta}{T}\,\mathrm{sinc}\left[\frac{(1-\beta)t}{T}\right] + \frac{\beta}{T}\,\mathrm{sinc}\left[\frac{\beta t}{T}+\frac{1}{4}\right]\cos\left[\frac{\pi t}{T}+\frac{\pi}{4}\right] + \frac{\beta}{T}\,\mathrm{sinc}\left[\frac{\beta t}{T}-\frac{1}{4}\right]\cos\left[\frac{\pi t}{T}-\frac{\pi}{4}\right] \quad (2.10)$$

and the corresponding frequency response as

$$F_{SRRC}(f) = \sqrt{F_{RC}(f)} \quad (2.11)$$

Both the raised cosine and square-root raised cosine pulse shapes are non-causal, and therefore, have infinite impulse responses. In practice, these pulse shapes must be truncated. The truncation is performed in a way that ensures their main lobes are preserved – that is, it must discard minimal pulse shaping information and must preserve the side lobes to some depth in the time domain.

2.4 Channel Models

The channel is the medium through which the signals travel to get from the transmitter to the receiver. We consider here two channel models rele-

vant to fixed-access microwave radio transmission, the additive white Gaussian noise channel and the multipath fading channel.

2.4.1 Additive White Gaussian Noise Channel

An ideal channel has no fading or other channel perturbations. Furthermore, the transmitter and receiver pulse shaping filters generate no ISI. The only concern for the receiver operating on an ideal channel is the disturbance caused by the presence of thermal noise. Most of this noise is produced internally by the receiver [14, 50]. Thermal baseband noise is approximated as AWGN because to a good approximation it is additive, white (at least over the signal band) and Gaussian. The noise is assumed to be wide-sense stationary. It has zero mean and a (two-sided) constant PSD as follows [13, 50]

$$\mathcal{S}_w(f) = \mathcal{F}\left[\frac{1}{2}E\left[w^*(t)w(t-\tau)\right]\right] = \mathcal{F}\left[N_0\delta(\tau)\right] = N_0; \quad -\infty \leq f \leq \infty \tag{2.12}$$

where $E[\bullet]$ denotes expected value; $\mathcal{F}[\bullet]$ is the Fourier transform with respect to τ; and $w(t)$ is the AWGN component.

An ideal channel is modelled using the AWGN channel model which typically results in the free space propagation model (cf. Figure 2.4). It is defined as having a clear, unobstructed path between the transmitter and receiver [48, 51]. Mathematically it is the simplest model and applies to a broad range of physical channels [56]. It is through the use of this channel model that many coding/modulation schemes initially have their performance quantified and compared.

The block diagram in Figure 2.5 shows the simple transmission model

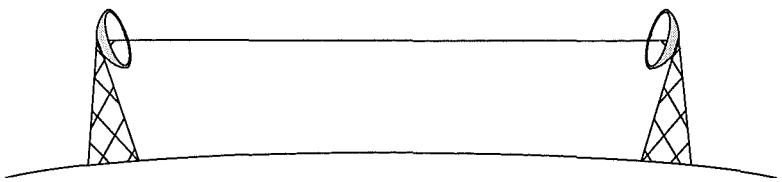


Figure 2.4: AWGN channel or free space propagation model

of the AWGN channel. Here the received signal, $r(t)$, is given as

$$r(t) = m(t) + w(t) \tag{2.13}$$

where $m(t)$ is the transmitted signal according to Equation (2.4). The white Gaussian noise component, $w(t)$, is simply superimposed (or added) to the signal and will alter the amplitude and/or phase of the received signal. Any two samples of AWGN are uncorrelated and statistically independent [50, 55].

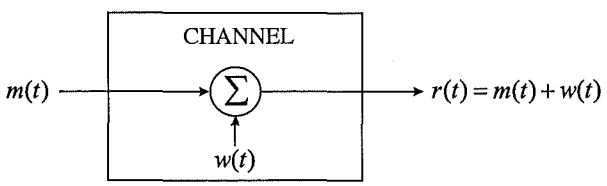


Figure 2.5: Block diagram of the AWGN channel.

2.4.2 Multipath Fading Channel

Many systems for wireless transmission operate in an environment inadequately described by the AWGN channel [58] – for example, the fixed-access line-of-sight digital microwave radio channel which is one example of a multipath fading channel [13, 14, 48, 50]. In such a channel, the received signal is the combination of components arriving via multiple channel paths (multipath propagation) reflecting or refracting off man-made or natural features within the transmission path – for example, the earth's surface, buildings, trees, and solar or atmospheric disturbances [14, 50, 59]. These multiple transmission paths produce more than one signal path propagating from the transmitter to the receiver (cf. Figure 2.6) [48]. It is not usually known *a priori* how many paths there will be or what delays individual paths will have.

The transmitter and receiver are stationary in fixed-access wireless sys-

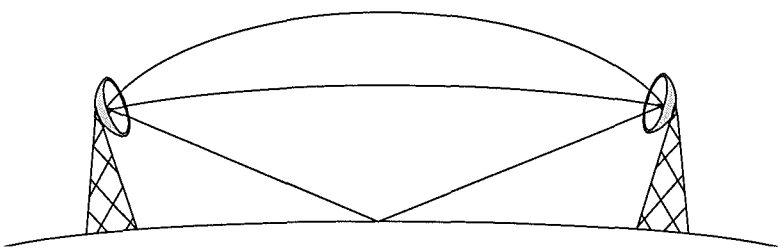


Figure 2.6: Fixed-access line-of-sight multipath fading channel model.

tems. This simplifies the situation as Doppler effects are not a significant issue as in mobile communications. As is typical in wireless access, we will assume that low gain (wide beam) antennas are used since they provide a more economically viable solution for small systems [60]. This makes for a worse case scenario in terms of system performance since the wide beam of the low gain antenna is more likely to "see" more multipath components than if high gain (narrow beam) antennas are used. Furthermore, as antenna gain is increased, system performance will improve.

Since we assume the use of low gain antennas there will probably be several fixed, "visible" scatterers or reflectors in the transmission path – for example buildings or trees. The resulting signal then consists of the weighted sum of a number of relatively delayed replicas of the transmitted signal due to the different lengths of transmission paths. These arrive at random times and introduce frequency-selective fading – that is, ISI [61]. Time-selective fading is due to time variations of scatters around the receiver. It is most apparent when the received signal strength is weak or near zero and/or has rapid phase changes [57].

The two most important types of fading observed for current practical systems are [51]: 1) time-selective, flat fading where all frequencies within the transmission band fade simultaneously with time – as for example, in the American IS-136 mobile system [51]; and 2) frequency-selective fading where some frequency components within the transmission band may be more attenuated than others and with only weak time variations – as for example, GSM [51]. Most of the fading experienced on the multipath fading channel un-

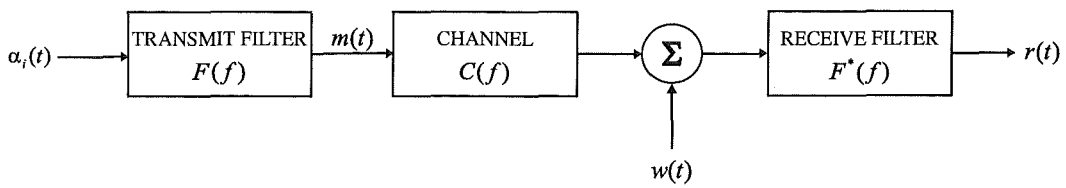


Figure 2.7: Transmitter and channel block diagram.

der consideration is frequency-selective fading with only weakly time-selective fading effects [48].

A slowly fading multipath channel introduces phase and amplitude distortions to the transmitted signal resulting in ISI [36]. The equivalent baseband frequency response, $C(f)$, of such a channel may be represented as [48, 62, 63]

$$C(f) = \sum_{\kappa=1}^{N_\kappa} c_\kappa e^{-j2\pi f \tau_\kappa} \tag{2.14}$$

where c_κ is a complex amplitude producing attenuation and a phase shift of each path; τ_κ is the relative delay of each path as seen at the receiver; and N_κ is the total number of paths. We assume that $\kappa = 1$ refers to the line-of-sight path. Its delay as seen by the receiver is defined as $\tau_1 = 0$. The delays of other paths are considered relative to the line-of-sight path. For simulation purposes the multipath fading channel is often modelled as a two path channel [48]. This is an adequate model in the narrow-band case as it is physically impossible to have more than one notch within the signal band.

Figure 2.7 show a block diagram of a system with a multipath fading channel. The overall frequency response, $H(f)$, of the transmitter and receiver pulse shapes and the multipath fading channel may be represented as

$$H(f) = F(f)C(f)F^*(f) \tag{2.15}$$

where $F(f)$ is the square-root raised cosine frequency response of the transmitter and receiver filters from Equation (2.11); and $C(f)$ is the frequency response of the channel from Equation (2.14).

The received complex baseband signal, $r(t)$, may then be written as

$$r(t) = \sum_i x_i h(t - iT) + n(t) \tag{2.16}$$

where x_i is the complex transmitted symbol; and $n(t) = w(t) \otimes f^*(-t)$ is the additive Gaussian noise at the output of the matched filter with variance of σ_n^2. The overall channel impulse response is

$$h(t) = f(t) \otimes c(t) \otimes f^*(-t) \tag{2.17}$$

where $h(t)$ is the impulse response of the transmit filter, $f(t)$, convolved with the impulse response of the channel, $c(t)$, convolved with the impulse response of the receiver matched filter, $f^*(-t)$.

2.4.3 Signal-to-Noise Ratio

The basis for comparing system performance is the signal-to-noise ratio (SNR) at the output of the receiver's matched filter. At any sampling time, i, of the matched filter, samples of the noise, $n(t)$, may be characterised by the

complex Gaussian probability function, $p\big(n(t)\big)$ [50, 54]

$$p\big(n(t)\big) = \frac{1}{2\pi\sigma_n^2}\exp\left(\frac{n_I^2(t) + n_Q^2(t)}{-2\sigma_n^2}\right) \tag{2.18}$$

The in-phase noise, $n_I(t)$, and the quadrature noise, $n_Q(t)$, are statistically independent.

From Equation (2.16), it may be seen that the noise power is limited by the receiver matched filter independently of the transmitted signal. The variance of the noise at the output of the matched filter is given as [50, 56]

$$\begin{aligned}
\sigma_n^2 &= \frac{1}{2}E\big[\big|n(t)\big|^2\big] \\
&= \frac{1}{2}E\big[\big|w(t) \otimes f^*(-t)\big|^2\big] \\
&= \int_{-\infty}^{\infty}\int \frac{1}{2}E\big[w(t_1)w^*(t_2)\big]f^*(t_1 - t)f(t_2 - t)dt_1dt_2 \\
&= \int_{-\infty}^{\infty}\int N_0\delta(t_1 - t_2)f^*(t_1 - t)f(t_2 - t)dt_1dt_2 \\
&= N_0\int_{-\infty}^{\infty}\big|f(t)\big|^2dt \\
&= N_0\int_{-\infty}^{\infty}\big|F(f)\big|^2df
\end{aligned} \tag{2.19}$$

where $F^*(f)$ is the frequency response of the matched filter pulse shape, which has been matched to the transmitter pulse shape. Since from Equation (2.6) $\int_{-\infty}^{\infty}\big|F(f)\big|^2df = \frac{1}{T}$ where T is the symbol period, the noise variance is

$$\sigma_n^2 = \frac{N_0}{T} \tag{2.20}$$

Taking the signal portion of Equation (2.16), the received symbol en-

ergy, E_s is given as

$$\begin{aligned} E_s &= E\Big[\int_{-\infty}^{\infty} \big|x_i h(t-iT)\big|^2 dt\Big] \\ &= E\big[|x_i|^2\big]\,\frac{1}{2}E\Big[\int_{-\infty}^{\infty} |h(t)|^2 dt\Big] \end{aligned} \tag{2.21}$$

Assuming $E\big[|x_i|^2\big] = 1$, the received symbol energy is then

$$E_s = \frac{1}{2}E\Big[\int_{-\infty}^{\infty} |h(t)|^2 dt\Big] \tag{2.22}$$

and the symbol power is $\frac{E_s}{T}$. The SNR calculated for the average received symbol energy (denoted here as SNR_0) is given by

$$\mathrm{SNR}_0 = 10\log_{10}\left(\frac{E_s}{N_0}\right)\mathrm{dB} \tag{2.23}$$

where N_0 is the noise PSD of the AWGN. SNR is also commonly expressed in terms of the average bit-energy-to-noise spectral density ratio and is given as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{E_b}{N_0}\right)\mathrm{dB} \tag{2.24}$$

Here, E_b is calculated as

$$E_b = \frac{E_s}{R\log_2 M} \tag{2.25}$$

where M is the number of signal points in the constellation; and R is the code rate – for an uncoded system $R = 1$.

2.5 Equalisation

In practice, we do not have prior knowledge of the channel characteristics. Furthermore, receivers often operate in wireless environments where

the channel is varying with time. Where equalisation is required, we employ an adaptive equalisation structure as it usually exhibits significantly better performance under varying conditions than fixed equalisers [64]. Adaptive equalisers may be linear, such as a transversal equaliser, or non-linear, as typified by the decision-feedback equaliser [37, 61, 65]. There are some disadvantages with linear equalisers – in particular, in coping with severe ISI (such as that due to a spectral null in the band); they often need a long filter length; and they can result in noise enhancement [13]. Here, we use decision-feedback equalisation (DFE) which effectively attempts to cancel much of the ISI in the received symbols. It operates by linearly reducing the ISI contributed by future symbols and cancelling the ISI contributed by past symbols using feedback of estimated symbols [13, 38].

As shown in Figure 2.8, a DFE usually consists of a forward transversal filter, a feedback transversal filter and a quantiser [37, 43]. The forward filter may either have symbol-spaced taps or fractionally-spaced taps [13, 37]. The forward filter acts as a linear equaliser to minimise the effect of pre-cursor ISI by using a weighted sum of the future received symbols. The feedback filter feeds back a weighted sum of previously detected symbols to cancel post-cursor ISI [38, 43]. Hence, only the effects of precursors have to be minimised by linear filtering. This results in less noise enhancement than when linear equalisation is employed. At time $t = iT$; $i = (0, 1, 2, \cdots)$, the quantiser uses the estimated, equalised symbol, z_i, to make a hard symbol decision, $\hat{x}_i$, by choosing the closest symbol from the given transmitted signal set.

The least-mean-square error (LMS) algorithm is often employed to

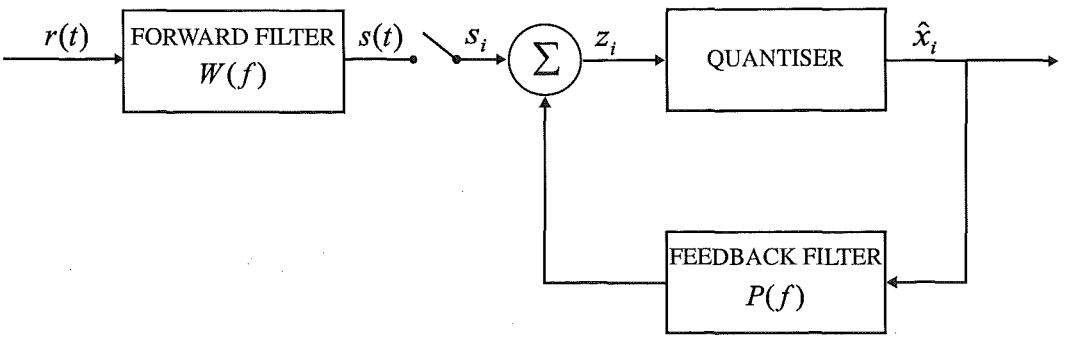


Figure 2.8: Block diagram of a decision-feedback equaliser.

adaptively update the DFE tap weights [37] because it is simple and convenient to implement [64]. The tap weights are initialised during a training period when known data are transmitted. This is used for feedback and to calculate the error signal of the LMS algorithm. Once the algorithm has converged, the DFE coefficients are assumed to be near optimum[2] [41]. The DFE then switches to a decision-directed adaptive mode for data transmission where the LMS algorithm adjusts the tap weights every symbol period to follow the slow variations of the channel response [13]. With reference to Figure 2.8, at time $t = iT$, based on the current output of the forward filter,

$$s_i = \mathbf{R}_i \mathbf{W}_i^H \tag{2.26}$$

and the previously detected symbols in the feedback filter, the unquantised

[2]In practice, optimality is not usually fully achieved during the training sequence. This becomes of interest when developing the combined equalisation and decoding structure in Chapter 5.

estimate of the symbol, z_i, in the i^{th} sampling interval is calculated by the DFE as

$$z_i = s_i + \hat{\mathbf{X}}_i \mathbf{P}_i^H \tag{2.27}$$

The vectors $\mathbf{R}_i = [r_{i+N_F-1}, \ldots, r_{i+1}, r_i]$ and $\hat{\mathbf{X}}_i = [\hat{x}_{i-N_B}, \ldots, \hat{x}_{i-2}, \hat{x}_{i-1}]$ are the samples at time i contained in the forward filter and the feedback filter respectively – where N_F and N_B are the numbers of taps in each filter; $\mathbf{W}_i$ and $\mathbf{P}_i$ are the row vectors of the complex conjugated forward and feedback tap weights at the i^{th} sampling interval.

The unquantised symbol estimate, z_i, is passed through the quantiser giving the quantised symbol, $\hat{x}_i$, at each symbol time. The quantiser is a hard-decision device making symbol-by-symbol decisions [64]. The residual error, e_i, is then estimated as

$$e_i = \hat{x}_i - z_i \tag{2.28}$$

The LMS algorithm uses a scaled version of e_i in the decision-directed adaptive mode to adapt and update the forward and feedback tap weights (at the $(i+1)^{th}$ sampling time) as

$$\mathbf{W}_{i+1} = \mathbf{W}_i + \mu e_i^* \mathbf{R}_i \tag{2.29}$$

$$\mathbf{P}_{i+1} = \mathbf{P}_i + \mu e_i^* \hat{\mathbf{X}}_i \tag{2.30}$$

where μ is a step size parameter chosen to ensure algorithm stability [37]; and e_i^* denotes the complex conjugate of e_i. The symbol values, $\hat{x}_i$, are assumed to be correct. They are shifted into the feedback filter of the equaliser and the

equalisation process is repeated for the next symbol at time $t = (i+1)T$ [38]. Equations (2.29) and (2.30) are also valid for the training mode, except that the residual error in Equation (2.28) is calculated using the known transmitted symbol.

2.6 Error Correcting Coding Principles

Two major communication system requirements demanded by present users are that they have extremely high data rates and outstanding data integrity. Furthermore, forecasts for the number of potential future users on the existing and increasingly crowded radio spectrum necessitate higher data rates than currently available. Error correcting codes (often referred to as channel codes) were developed to enhance the immunity of the transmitted data to various channel impairments such as noise and fading [23]. Historically there has been both an academic and a commercial drive towards the perfection of error correction codes.

Error correcting codes add (controlled) redundant information to the transmitted binary data sequence. This allows the receiver to detect and, in some cases, correct errors [23]. The redundant information is usually added in such a way that an appropriate trade-off between code performance and overall code rate is made.

The actual throughput (in bits/second) at the output of the channel encoder is given as [50]

$$R_o = \frac{k}{n} R_s \tag{2.31}$$

where R_s is the channel rate in bits per second; k is the number of original data bits transmitted; and n is the total number of bits per block in the data sequence after redundancy has been added. The code rate, R is a dimensionless ratio and is expressed as

$$R = \frac{k}{n} \tag{2.32}$$

Effectively, the controlled redundancy added by the code is such that the encoded data has a certain dependency among some or many symbols. This enables the receiver to detect the incoming data more accurately [66]. The drawback of channel coding is that the overall data rate is reduced. To achieve the same data rate as an uncoded system, one may either increase the signalling rate if the bandwidth can be expanded or increase the number of bits per symbol if the channel is band-limited.

There are two distinct types of channel codes: error detection codes; and error detection and correction codes. Error detection codes are designed to detect incoming errors but with no, or unused, error correction capability. In some receivers where data integrity is not crucial – for example, some voice applications – errored words are unable to be retransmitted due to system delay constraints. In these cases the word in error is simply processed with the error, or the errored word may be muted whereby it is set to some predefined value in order that transmission continues effectively delay-free. In most situations, the message is able to be understood satisfactorily [23]. In these applications, error detection is employed to check that there are not too many errors over a long period of time – that is, that the channel is not too corrupt. In other applications – for example data transmission – on the detection of

an error, the receiver sends an automatic repeat request (ARQ) requesting the retransmission of the erroneous word. ARQ is typically applied in data communications where the delay of retransmission is not a critical issue but very high data integrity is required.

For some information sources – for example, high capacity microwave radio systems – data muting does not provide high enough data integrity or the delay produced by ARQ is too large. In such cases error detection and correction coding may be used. The QoS required by a system and the information (voice, video or data) being sent via the channel are the usual criteria by which to choose what scheme to utilise. One method of choosing the combination of error detection and/or correction to use is to categorise the QoS such that service with zero probability of loss is at one extreme and service with minimal delay is at the other extreme. The first is best served with ARQ, the latter with error detection/correction schemes and the remainder using either one or a combination of the two [67].

Codes performing both error detection and correction are commonly referred to as forward error correcting (FEC) codes. They determine from a predefined set of possible code words what data was originally sent [23]. Forward error correction codes generally fall into two discrete categories: block codes and convolutional codes. There is a basic difference between the two coding strategies. Block coding encodes a finite number of bits into each coded block of data [23] and each block is independent of any others. Therefore, a block code is a memoryless form of coding.

On the other hand, a convolutional encoder may be viewed as a finite-

state machine or a finite impulse response (FIR) digital filter [23,50], whereby a continuous sequence of binary data is passed through a linear shift register (convolutional encoder) to form a continuous code word [13,68]. The output of the convolutional encoder at the current time step is a function of the input bits and the existing encoder contents. The encoder is said to be a finite memory system where its memory is defined by the contents of the shift register [36].

A disadvantage of a convolutional code is that if its memory is large, the complexity of the decoder is also large. At a given level of complexity, the performance of a convolutional code is often better than a block code. However, block codes may be more easily algebraically decoded (hard-decision decoded) than large memory convolutional codes. Since a convolutional code converts the entire data stream into a single code word, there can be a variable decoding delay. The per block decoding delay for a block code is consistent for the code, which can be advantageous in applications – for example, voice – where decoding delay is an important issue. Finally, some protocols transmit blocks of digital data; therefore, block codes are more efficient than convolutional codes since they do not require truncation between data blocks.

Code performance is often measured in terms of coding gain. Coding gain is a measure of the difference in SNR at a specified bit error rate (BER) between the performance of an uncoded and a coded system. To measure the performance of coding schemes it is usual to compare systems with similar spectral efficiencies, η, or throughput, ρ. One may also compare the error probabilities, P_e, of different schemes versus their average bit-energy-to-noise ratio, $\frac{E_b}{N_0}$.

2.7 Linear Block Codes

In this Thesis we concentrate on the application of a hierarchy of linear block codes to form coded modulation schemes. Linear block codes are the most easily implemented form of block codes and are most often used [23, 66].

A linear block code, C, has a finite number of possible code words. Block coding takes a number of input data bits, k, and calculates a group of r check bits to give a total of n bits to transmit. The check bits applied by the code are usually derived from the input data bits using a predefined generator matrix, $\mathbf{G}$. The generator matrix is made up of a set of k linearly independent vectors, $\mathbf{G} = [\mathbf{G}_1^T, \mathbf{G}_2^T, \ldots, \mathbf{G}_k^T]^T$, where k is the size of the transmitted input data block. Each generator vector, $\mathbf{G}_j$, has length n, where n is the total number of bits to be transmitted. Thus we can define a generator matrix as a k x n array [55]:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ \vdots \\ \mathbf{G}_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & & & \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix} \tag{2.33}$$

The generator matrix of some codes may be put in the form

$$\begin{aligned} \mathbf{G} &= \left[\begin{array}{c|c} \mathbf{P} & \mathbf{I}_k \end{array} \right] \\ &= \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & \vdots & \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix} \end{aligned} \tag{2.34}$$

where $\mathbf{I}_k$ is a k dimensional identity matrix; and $\mathbf{P}$ is the parity array portion of the generator matrix. The resulting codes transmit the original data as a block followed by a block of parity bits or symbols, this is referred to as systematic encoding. The advantage of using systematic encoding is that decoding complexity is simplified since it is not necessary to store the identity matrix portion of the generator matrix [23, 53, 55]. It can also make data decisions with no decoding.

If we have a sequence of k uncoded bits, given as a row vector

$$\mathbf{a} = [a_1, a_2, \ldots, a_k] \tag{2.35}$$

the generator matrix directly encodes these uncoded message bits such that

$$\mathbf{c} = \mathbf{aG} = a_1\mathbf{G}_1 + a_2\mathbf{G}_2 + \cdots + a_k\mathbf{G}_k \tag{2.36}$$

The code vector, $\mathbf{c}$, is now a linear combination of the generator matrix corresponding to the original sequence of uncoded bits [23]. In binary block coding, the arithmetic in Equation (2.36) is modulo-2. The set of possible code words is completely defined by the generator matrix. This enables a reduction in complexity of both the encoder and decoder since they only need to store the contents of the generator matrix to encode and decode the data [23, 55].

A block code may either be soft- or hard-decision decoded. Hard-decision decoding operates on the hard-decision at the output of a quantiser and algebraically decodes the received symbols on a symbol-by-symbol basis. The hard-decision decoded code word may be reconciled into its original information bits in various ways. Two common methods are: the use of a simple table look-up, which is sufficient for a code with a small number of code words;

or the use of the parity check matrix, $\mathbf{H}$, to form a syndrome, $\mathbf{s}$, as noted below. The latter method is more computationally efficient for codes with a large number of code words.

The parity-check matrix, $\mathbf{H}$, has the property that $\mathbf{G}\mathbf{H}^T = \mathbf{0}$. It is used to indicate if the received code vector, $\mathbf{r}$, is a valid member of the set of possible code words. The received code vector may be regarded as the (modulo-2) summation of the transmitted code vector, $\mathbf{c}$, and the error pattern, $\mathbf{e}$, produced by noise, such that [23]

$$\mathbf{r} = \mathbf{c} + \mathbf{e} \tag{2.37}$$

Depending on the error correction capability of the code, the parity-check matrix can be used to indicate the validity of the received code word, as well as detect and/or correct some of the bits in error. This is achieved by using the syndrome vector of $\mathbf{r}$ which is defined as [23, 55]

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T \tag{2.38}$$

If the syndrome vector $\mathbf{s} = \mathbf{0}$, then $\mathbf{r}$ is a valid code word. The syndrome vector is independent of the transmitted code word, $\mathbf{c}$. From Equations (2.37) and (2.38), it may be shown that the syndrome vector is solely a function of the error pattern as follows [23, 55]

$$\begin{aligned} \mathbf{s} = \mathbf{r}\mathbf{H}^T &= (\mathbf{c} + \mathbf{e})\mathbf{H}^T \\ &= \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T \\ &= \mathbf{0} + \mathbf{e}\mathbf{H}^T \\ &= \mathbf{e}\mathbf{H}^T \end{aligned} \tag{2.39}$$

If an error has occurred in the received code word and it is detectable, $\mathbf{s} \neq \mathbf{0}$. If the errors in $\mathbf{r}$ are correctable, the syndrome vector identifies the error and is used to correct for the particular error pattern [23, 55].

By contrast, soft-decision decoding considers a sequence of symbol estimates and their "side", or soft, information detailing how close each received symbol was to its decision boundary. The soft information is formed by quantising the output of the receiver to multi-bit words – effectively, the soft-decision decoder would ideally operate on an analogue output but this is not achievable. Using this soft information, the Euclidean distance between the estimated and the received symbol is formed. This extra information allows the decoder to make decisions on the received data with higher reliability than hard-decision decoding. When soft-decision decoding is implemented on standard block or convolutional codes there is approximately a 2-dB SNR advantage over hard-decision decoding [13].

Soft-decision decoding algorithms fall into two basic categories: optimum and sub-optimum. Two examples of optimum soft-decision decoding algorithms are: maximum likelihood sequence estimation (maximum likelihood decoder); and maximum *a posteriori* probability (MAP) symbol-by-symbol decoder [13].

The Viterbi decoder, which is optimum in a sequence sense [66, 69] is an example of maximum likelihood decoding. The Viterbi decoder is a trellis-based scheme whereby hypothesised sequences of symbols are compared by way of a metric within a trellis structure to obtain the most likely sequence of symbols sent. A maximum likelihood decoder makes a decision on the received

code word, $\mathbf{r}$, given a set of possible code words, $\mathbf{c}$ – that is, $p(\mathbf{r}|\mathbf{c})$ [23]. It is constructed using the theorem of non-optimality [66], whereby nodes in the trellis corresponding to identical states are merged and the most likely path is retained for each state. This allows the construction of a decoder with significantly less complexity than an exhaustive tree search [66].

MAP symbol-by-symbol decoding computes a set of *a posteriori* probabilities to determine the most likely symbol sent. MAP decoding effectively derives the probability that a bit in a code word, c_k, is transmitted conditioned on the received code word, $\mathbf{r}$ – that is, $p(c_k|\mathbf{r})$ [23]. The complexity of the MAP decoder is higher than that of maximum likelihood decoding. However, a useful by-product is that the *a posteriori* bit probabilities and channel symbols are produced which may be used in applications such as turbo decoding [66]. In many applications the performances of the maximum likelihood decoder and the MAP decoder are equal [66].

The sub-optimum algorithms may be based on maximum likelihood decoding or MAP decoding. For example: reduced-state sequence estimation (RSSE) [70]; schemes based on maximum-likelihood hard-decision decoding, such as Wagner decoding [71]; the Chase algorithm [72]; or other schemes based on the Wagner decoding principles which are forms of soft-decision decoding.

In this Thesis we focus on Viterbi decoding which yields maximum likelihood decoding in a sequence sense [68]. It was first proposed in 1967 [26] and was intended for decoding convolutional codes [15]. It may also be applied to block codes. The major difference between the block code trellis and the convolutional code trellis is that the block code has only one state at the

beginning and conclusion of each block – that is, it has a periodically time varying trellis structure. On the other hand, the convolutional code trellis has one state at the start of the code and then many branches until it is terminated by a long run of ones or zeros.

The Viterbi Algorithm is well suited for block coding (providing the number of states is not too large as it becomes excessively complicated to decode) since a block code defines a sequence of symbols in each block and trellis decoding is the optimum algorithm for decoding sequences. However, in general, it has a high decoding complexity due to the code having many states. The method for decoding the block code trellis is similar to that of a convolutional code. Whereby the minimum distance of each path merging into a state is calculated keeping the path with the lowest minimum distance and discarding all others. This continues until the end of the block code and the one path left in the trellis is considered the correctly decoded path.

The metric commonly used to describe the error correcting capability of a linear block code is the minimum Hamming distance, d_{min} [73]. The Hamming distance describes the number of coordinates, or bits, in which a pair of code words differ. The minimum Hamming distance between any two code words of a code gives an indication of the code's error detection and correction capability [23]. A code with a minimum Hamming distance of d_{min} can detect all error patterns of weight less than or equal to $(d_{min} - 1)$ and correct all error patterns of weight less than or equal to $\lfloor (d_{min} - 1)/2 \rfloor$ [23].

A block code, C, may be compactly represented using the notation

$$C = (n, k, d_{min}) \tag{2.40}$$

where n is the encoded block length; k the number of data bits per block.

2.7.1 Reed-Muller Codes

The research done by Muller [22] followed by that of Reed [21] developed a particular error-correcting code construction. This combined work resulted in a family of linear binary block codes called "Reed-Muller" codes [23, 45]. They were introduced at the first International Symposium on Information Theory in Cambridge, Massachusetts, 1954 [16]. Reed-Muller codes are one of the oldest families of codes, after codes such as the Hamming codes and the Golay code. Reed-Muller codes have played a large role in the history of coding as they are easily encoded and decoded and form a large family of simple codes. Additionally, they are relatively flexible in correcting a varying number of errors per code word [23, 45] and may easily be soft-decision decoded since they have a simple trellis structure [74]. A drawback associated with many Reed-Muller codes is having long runs of zeros and ones as the code word.

Among other applications in the early days of coding, Reed-Muller codes played a significant role in error control in the deep space probe research programs run by the United States of America – for example, they were used in the Mariner-class space-craft [23, 45]. Interest in them waned in the late 1970's and 1980's due to advances in other coding techniques such as BCH codes in 1959 [23] and Reed-Solomon codes in 1960 [16]. However, they are now enjoying renewed interest due to their flexibility in correcting a varying number of errors per code word and their extremely fast soft- and hard-decision decoding techniques [23].

Reed-Muller codes are denoted $\mathcal{R}(r, m)$, where $m = (2, 3, \dots)$ and defines the encoded block length equal to 2^m; and r is the code order such that $0 \leq r \leq m$. The properties of Reed-Muller codes for any m or order r are given in Table 2.1 [23, 30, 45].

Table 2.1: Properties of Reed-Muller codes.

block length	$n = 2^m$
minimum distance	$d_{min} = 2^{m-r}$
data block length and dimension	$k = 1 + \sum_{l=1}^{r} \binom{m}{l}$

Using these properties, Table 2.2 gives a collection of Reed-Muller codes showing their encoded word length, number of data bits and minimum Hamming distance [23].

Table 2.2: Reed-Muller codes expressed as $C = (n, k, d_{min})$.

	$m =$ 2	3	4	5
r =0	(4,1,4)	(8,1,8)	(16,1,16)	(32,1,32)
1	(4,3,2)	(8,4,4)	(16,5,8)	(32,6,16)
2		(8,7,2)	(16,11,4)	(32,16,8)
3			(16,15,2)	(32,26,4)
4				(32,31,2)

The generator matrices to encode Reed-Muller codes are easily con-

structed from a set of linearly independent basis vectors [23,45]. The generator matrix associated with each binary Reed-Muller code consists of vectors which are defined by all Boolean functions of degree less than or equal to the code order, r, in m variables [23,45]. As an example, the basis vectors for the second order Reed-Muller code of length 16, $\mathcal{R}(2,4)$, are given by the monomials in four variables of degree two or less as [23,45]

$$1, v_1, v_2, v_3, v_4, v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4, v_3v_4 \tag{2.41}$$

The binary basis vectors associated with Equation (2.41) are then

$$\begin{aligned}
\mathbf{1} &= (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)\\
\mathbf{v}_4 &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)\\
\mathbf{v}_3 &= (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)\\
\mathbf{v}_2 &= (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)\\
\mathbf{v}_1 &= (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)\\
\mathbf{v}_3\mathbf{v}_4 &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)\\
\mathbf{v}_2\mathbf{v}_4 &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)\\
\mathbf{v}_1\mathbf{v}_4 &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)\\
\mathbf{v}_2\mathbf{v}_3 &= (0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1)\\
\mathbf{v}_1\mathbf{v}_3 &= (0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1)\\
\mathbf{v}_1\mathbf{v}_2 &= (0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)
\end{aligned} \tag{2.42}$$

These basis vectors are employed (in the order shown) as rows in the generator matrix for the $\mathcal{R}(2,4)$ Reed-Muller code, which is the (16,11,4) code of Table 2.2. Note, that the Reed-Muller codes include the extended Ham-

ming codes – shown in Table 2.2 are the extended Hamming codes $(8,4,4)$, $(16,11,4)$ and $(32,26,4)$ which are extended from the $(7,4,3)$, $(15,11,3)$ and $(31,26,3)$ Hamming codes respectively.

Since Reed-Muller codes are linear codes, their properties and structure are inherently easy to analyse [23]. The (Hamming) weight, ω, of a code word is defined as the number of nonzero elements it contains. One property of a linear code is that its minimum Hamming distance, d_{min}, is equal to the lowest weighted nonzero code word [23,45]. The weight distribution, A_ω, of a code is defined as the number of code words of weight ω. $A_{\omega=d_{min}}$ is referred to as the error coefficient of a code which is the number of nearest neighbour code words at the minimum Hamming distance. The error coefficient of a code, $A_{d_{min}}$, is used when determining the analytical approximation to its performance and for Reed-Muller codes is given by [30,45]

$$A_{d_{min}} = 2^r \cdot \prod_{l=0}^{m-r-1} \frac{2^{m-l}-1}{2^{m-r-l}-1} \tag{2.43}$$

where r is the order of the code; 2^m is the block length; and $d_{min} = 2^{m-r}$ is the minimum Hamming distance.

2.7.2 Decoding Reed-Muller Codes

Reed-Muller codes may be hard- or soft-decision decoded to recover the transmitted information sequence. The hard-decision decoding algorithm used in this Thesis and commonly used in other applications is the Reed decoding algorithm [23,45]. The Reed decoding algorithm uses majority logic decoding and provides maximum likelihood hard-decision decoding in an efficient

manner [23, 45] – worked examples of the Reed decoding algorithm are given in [23, 45]. Essentially, it is implemented by forming a number of orthogonal check sums for each message bit. This is done by using a combination of independent elements of the received signal for each check sum. Each check sum is calculated and if they all give the same result, it is assumed no errors have occurred. In this case, each check sum shows the correct value of the message bit in question. If one error has occurred, exactly one of the check sums will differ from the majority of check sums. In this case the error can be corrected by choosing the outcome of the majority of the check sums, and so on for additional errors. As soon as a check sum majority no longer exists, the decoder can no longer correct the errors. The message bits associated with the highest order basis vectors in the generator matrix (cf. Equation (2.42), basis vectors $\mathbf{v}_1\mathbf{v}_2$, $\mathbf{v}_1\mathbf{v}_3$ and so on) are checked first, progressing towards the lowest order basis vector (cf. Equation (2.42), basis vector $\mathbf{1}$). The Reed decoding algorithm may be used to decode Reed-Muller codes of any order [23, 45].

The soft-decision decoder often used for Reed-Muller block codes is a trellis decoder implemented using the Viterbi Algorithm. Due to their consistent structures, the advantage of implementing trellis decoding on block codes is that they lend themselves to efficient and predictable decoder structures – that is, their decoding delay and complexity remains constant. A new trellis is started at the beginning of each block and terminated at the block's conclusion.

Figure 2.9 shows the four-section trellis construction of the Reed-Muller code $\mathcal{R}(1, 3)$ and Figure 2.10 shows the four-section trellis constructions of the

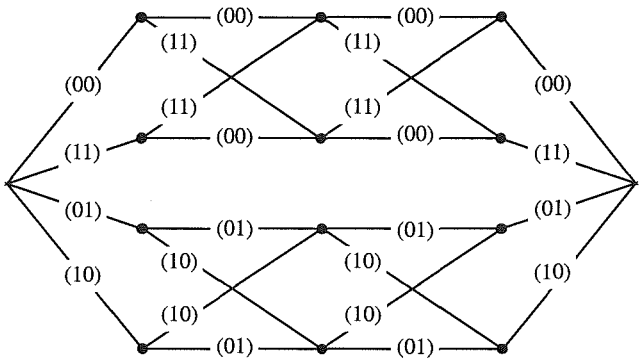
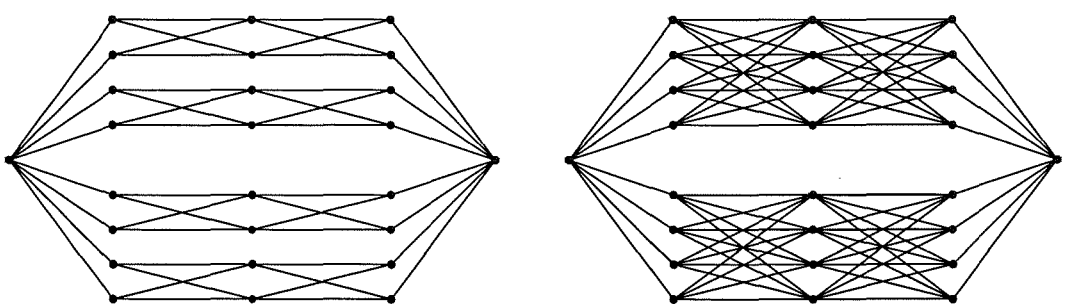


Figure 2.9: Trellis construction of the Reed-Muller block code $\mathcal{R}(1,3)$.



(a) $\mathcal{R}(1,4)$ (b) $\mathcal{R}(2,4)$

Figure 2.10: Trellis constructions of two Reed-Muller block codes.

$\mathcal{R}(1,4)$ code and the $\mathcal{R}(2,4)$ code. From Table 2.2 these are the $(8,4,4)$, $(16,5,8)$ and $(16,11,4)$ Reed-Muller codes respectively.

Every (block) code word has a unique path through the trellis [23]. The trellis is constructed by using each bit from each code word as a sequence of branch labels separated by nodes. As shown in Figure 2.9 for the $\mathcal{R}(1,3)$ code, if there are two labels resulting in a node with only one branch entering and exiting, they are concatenated. This results in a unique branch labelled by two bits - for example, (00), (11), (01) or (10) - entering and exiting each node and reduces the decoder complexity. At each node, the Euclidean distance between the received word and the trellis paths incident on each state are calculated. The path with the smallest Euclidean distance is retained. Using this procedure, the maximum likelihood path through the trellis is determined at the conclusion of decoding. As the length of the Reed-Muller code increases, more bits are concatenated onto each path where appropriate – for example, the trellises in Figure 2.10 have four states in each branch. As may be seen, trellis constructions for Reed-Muller codes of similar block lengths have the same number of states. However, the complexity of the internal computations increases as the code order increases – since, as the length of the code increases, the number of states in the trellis also increases [66].

2.8 Coded Modulation

As discussed in Section 2.6, when adding an error correcting code to a system the overall information rate is reduced. There are two approaches to maintaining a given data rate - either to expand the transmission bandwidth

or to increase the modulation efficiency (bits per symbol). For a band-limited channel, a given data rate may be obtained after coding by increasing the number of signal points in the constellation – for example, increasing the constellation size from 16-QAM to 32-QAM. If the coding and modulation are designed independently as was done conventionally, the full advantage of increasing the constellation size is not properly realised unless extremely powerful codes are used to offset the performance loss due to signal set expansion and to provide some coding gain [13]. These powerful codes are generally large constraint convolutional codes or block codes with a large block length, both of which translate to a large decoding complexity and delay [13]. However, by combining the design of coding and modulation, referred to as coded modulation, we may conserve bandwidth while saving power. These schemes often use codes which are relatively simple to encode and decode [14]. The original coded modulation scheme was developed by Ungerboeck in 1982 [28] and is referred to as 'trellis-coded modulation' (TCM).

TCM is based on the basic tenet that given a signal constellation – for example, 8-level phase-shift keying (8-PSK) or 16-QAM – the most likely error a decoder will make is between neighbouring signal points. Instead of transmitting each symbol independently, TCM transmits a sequence of symbols. Furthermore, by partitioning the signal set using Ungerboeck's set partitioning methods (to be described in Section 2.8.1) the minimum squared Euclidean distance (MSED) between valid sequences of symbols is also increased. The sequence of allowed symbols is usually defined by a convolutional code within the Ungerboeck encoder. The decoding of TCM is usually implemented by means

of the Viterbi Algorithm [75]. The joint coding, constellation and signal mapping functions of TCM are aimed at achieving the maximum free distance of the code and thereby maximising its performance while maintaining a constant transmission bandwidth. The free distance of the code is usually defined by the MSED between the code's parallel transitions [13]. Parallel transitions are a result of having uncoded bits in the last subset level resulting in more than one signal point per subset being transmitted.

TCM may be performed using convolutional codes of any rate, however, the preferred code rate is $R = \frac{k}{k+1}$ – that is, only one redundant bit in every $k+1$ transmitted bits [28]. This requires a doubling of the signal constellation size. However, as the rate and constraint length of the code increases, the trellis decoding complexity and delay becomes large. An alternative coded modulation scheme may be designed by replacing the single convolutional code with a number of less complex codes, one assigned to encode each partition level of the constellation. This form of coded modulation is referred to as multilevel coding. Multilevel coding also restricts the transmitted signals to certain signal sequences. Its overall code structure is complex, however, each component code is usually easily decoded thereby overcoming some of the inherent complexity issues of some TCM schemes.

2.8.1 Ungerboeck Set Partitioning

Ungerboeck's mapping by set partitioning is one technique that may be used to assign bit labels to signal constellations [28]. It is aimed at increasing the MSED between signals in each partitioned subset in order to increase

MSED between sequences of coded points. Mapping by set partitioning was originally developed and used in TCM [76]. It may however also be applied successfully to other coding techniques. It is based on partitioning a basic signal constellation, S_0, into the subset chain $S_0/S_1/\cdots/S_m$, where m is the number of times S_0 is partitioned [28]. The partitioning of S_0 into its subsets S_1 is denoted S_0/S_1. In general, a signal set may be partitioned 2^n ways, where $n = (1, 2, 3, \cdots)$. The most common partitioning is when $n = 1$, which is used in binary partitioning. The subsets in each partition level are usually chosen to have an equal number of signal points. Subsets in successive levels of the partition chain usually are designed to have an increasing minimum Euclidean distance between signal points [11] such that

$$\delta_0 \leq \delta_1 \leq \cdots \leq \delta_m \tag{2.44}$$

where δ_l^2, $l = (0, \cdots, m)$, is the MSED between elements in a partitioned signal subset, S_l, and is also the inter subset distance at partition level $(l+1)$ (for subsets with the same immediate parent subset) resulting from further partitioning of S_l. If the MSED is not increasing, coding schemes based on set partitioning may not reach their full potential.

Figure 2.11 shows one possible example for binary set partitioning of 16-QAM. Each symbol label in Figure 2.11 may be denoted $(x_4\, x_3\, x_2\, x_1)$ – where x_4 is the label's most significant bit (MSB) and x_1 its least significant bit (LSB). To map bit labels to signal points we start with the first partition S_0/S_1, where S_1 has two subsets with a larger MSED than S_0. We label the LSB, x_1, of one S_1 subset with a zero and the LSB of the other with a one. The two subsets of S_1 are then each partitioned via the partition chain S_1/S_2

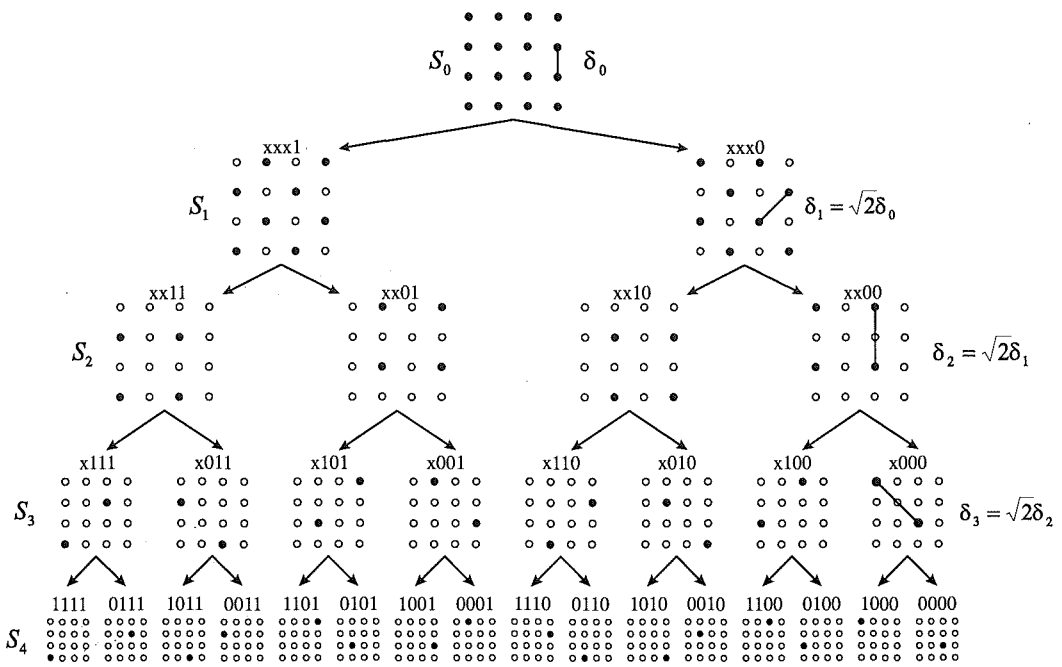


Figure 2.11: Set partitioning for a two-dimensional constellation.

into two more subsets giving four subsets at subset level S_2. In each subset the bit labels, x_2, are assigned a similar mapping of ones and zeros. This subset partitioning may be continued until all bits of the label have been assigned (cf. Figure 2.11). It is often sufficient for large constellations to be partitioned two or three times leaving the last signal subsets with more than one signal point.

By combining mapping by set partitioning with an error correcting code, we may restrict the sequence of symbols sent to a predefined set of sequences [50]. Block or convolutional codes may be associated with signal set labelling via set partitioning. An error correcting code may choose a sequence of labelled

signal points or it may be used to encode a bit sequence at each partition level. The latter is the philosophy of multilevel codes as will become clear in Chapter 3.

2.8.2 Multidimensional Signal Constellations

There are several methods of constructing multidimensional constellations. The method used in this Thesis is to take the Cartesian product, or concatenation, of two or more two-dimensional signal constellations and treating it as one multidimensional constellation. Multidimensional constellations are another example of where coded modulation may be employed [44]. Again, the coding restricts the sequence of signal point combinations. For example, restricting the use of the outer-most symbols may reduce the effect of distortion by amplifier non-linearities and thereby increase system performance. This will also reduce the peak-to-average power ratio (PAR) of the transmitted signal. Multidimensional codes may also be employed in conjunction with multilevel coding.

High-capacity, multidimensional signal constellations may often be constructed as subsets of lattices [44]. The simplest of these is the integer lattice, Z^{2N}, from which a $2N$-dimensional constellation may be constructed as a sequence of N two-dimensional constituent QAM constellations [46]. Given a two-dimensional constellation S_0 from Z^2, a four-dimensional constellation may be created by concatenating S_0 in the form $S_0 X S_0$, as the Cartesian product of S_0 with itself. Effectively, each signal point in S_0 is combined with every point in S_0 resulting in uniquely labelled four-dimensional signal points.

Multidimensional constellations provide a natural extension to current QAM system designs. A multidimensional coded constellation requires a smaller constituent constellation for a given data rate than a two-dimensional QAM constellation as the coding redundancy may be spread over all dimensions.

A multidimensional constellation in Z^{2N} may also be set-partitioned, although this may not work for all lattices. This is achieved by partitioning its constituent two-dimensional constellation, S_0, to obtain the subset chain $S_0/S_1/\cdots/S_m$, where m is the number of partitions [28, 44]. The subsets at each level are usually chosen to have an equal number of points. Here, the MSED inside the subsets of each partition level should also be increasing as in Equation (2.44). It is assumed for convenience that the constituent constellation, S_0, has unity average power. Once the constituent constellation has been partitioned, each partition is concatenated N times to form the $2N$-dimensional constellation partitioning [44]. Figure 2.12 illustrates the partitioning and concatenation to form a four-dimensional constellation. In Figure 2.12, A_0 corresponds to the original constituent constellation S_0, B_i corresponds to the subsets in the first partition S_1 and C_i corresponds to the subsets in the partition S_2. This labelling format continues through each of the partition levels. A multidimensional constellation set partitioned in this manner has the same MSED between signal points within a subset as the subsets of the associated partition in the constituent constellation.

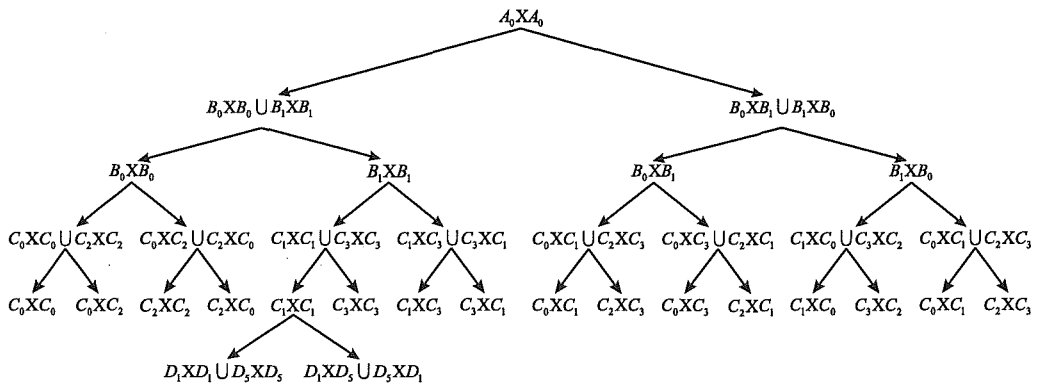


Figure 2.12: Set partitioning for a four-dimensional constellation.

2.8.3 Shaping Codes for Multidimensional Constellations

Shaping codes are a convenient method of preparing multidimensional signal points for transmission. They may also be used to reduce the average PAR of a multidimensional signal constellation. Consider the transmission of $(NQ+1)$ bits per $2N$-dimensional signal point. This requires a $2N$-dimensional constellation of 2^{NQ+1} points [44]. Each constituent constellation S_0 is chosen to contain $M = 2^Q + \frac{2^Q}{N}$ points arranged as 2^Q inner points (standard QAM) and $\frac{2^Q}{N}$ outer points. The outer points are placed symmetrically around the inner points such that the PAR is constrained [44,77]. The Cartesian product of N such constellations forms the $2N$-dimensional constellation.

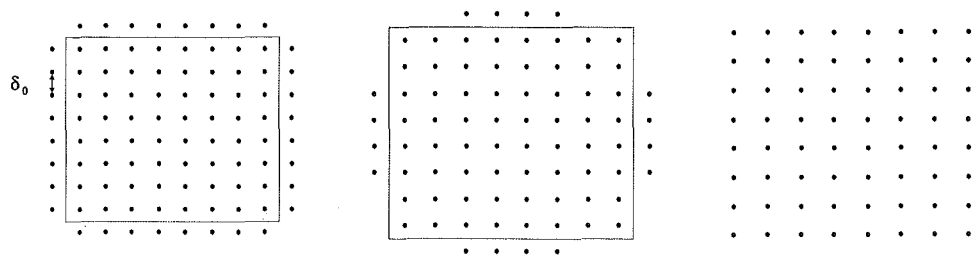
Multidimensional signal points containing more than one outer point from its constituent constellation are excluded by the encoding process. This

is referred to as a shaping code [44, 78]. Given a two-dimensional constellation containing $M = 2^Q + \frac{2^Q}{N}$ points, this restricts the number of points in the resulting multidimensional constellation to 2^{NQ+1} and reduces the average power in each constituent constellation to $\frac{2N-1}{2N}$ times the average power of the inner points plus $\frac{1}{2N}$ times the average power of the outer points [44].

When $(NQ+1)$ bits are mapped to a $2N$-dimensional signal point using a shaping code, the MSB is used to indicate whether an outer point is to be selected. As an example, a MSB equal to one indicates that an outer point is to be included (used), the next $\log_2 N$ bits determine its location in the sequence of N two-dimensional points. The next $(N-1)Q$ bits select the $(N-1)$ inner points and the remaining $(Q - \log_2 N)$ bits select the single outer point. If there is no outer point the MSB will equal zero, the next NQ bits choose the N inner points [44].

The M points in S_0 are labelled using $(Q+1)$ bits. To assign bit labels to a constituent constellation it is partitioned using the methods described in Section 2.8.1. Again, the subsets at each level are chosen to have an equal number of points and, in particular, an equal number of outer points. The 2^Q inner points are uniquely labelled via set partitioning using the Q low order bits with the MSB, or $(Q+1)^{st}$ bit, set to zero. The $\frac{2^Q}{N}$ outer points are labelled via set partitioning by the $(Q + 1 - \log_2 2N)$ low order bits with the remaining $\log_2 2N$ high order bits set to one.

In the latter work, we use 96-QAM and 80-QAM signal constellations as examples. For both these constellations, their associated two-dimensional constituent constellation, S_0, and distribution of inner and outer points are



(a) 96-QAM (b) 80-QAM (c) 64-QAM

Figure 2.13: Two-dimensional constituent constellations, S_0, showing the boundary between inner and outer points and the minimum Euclidean distance, δ_0, between adjacent signal points.

illustrated in Figure 2.13. The multidimensional labelling is easily achieved by labelling S_0. In both these cases the $2^Q = 64$ inner points (standard 64-QAM – cf. Figure 2.13(c)) are labelled $0xxxxxx$, where x indicates the bit labels selected via set partitioning. In the 96-QAM case the $\frac{2^Q}{N} = 32$ outer points are labelled $11xxxxx$ and for 80-QAM the $\frac{2^Q}{N} = 16$ outer points are labelled $111xxxx$. Once the constituent constellations have had their signal points labelled, the Cartesian product of N of these constellations forms the $2N$-dimensional constellation – for example, in the 80-QAM case the eight-dimensional constellation is formed as $S_0XS_0XS_0XS_0$ with multidimensional signal points containing more than one two-dimensional outer point excluded. Such constellations lead to minimal signal set expansion for any desired com-

bination of codes in a multilevel structure [44].

2.9 Literature Review

As already outlined in Section 1.4, this Thesis makes original contributions in two major areas: 1) multilevel block code design and their performance approximation; and 2) combined equalisation and multilevel block decoding. Here we provide a brief literature review discussing some of the previous work relevant to these two areas of interest.

2.9.1 Multilevel Coding Techniques

Imai and Hirakawa in 1977 [27] first proposed multilevel coding methods based on a hierarchy of binary block codes. Cusack [79] and Sayegh [80] later extended this concept to include QAM constellations. Ginzburg [81] proposed multilevel coding techniques that allow more elaborate set partitioning regimes than originally suggested. Tanner [82] extended Ginzburg's proposal to formally link the minimum Euclidean distance of the set partitions in a multilevel code with the minimum Hamming distance of each component code. This work was initially viewed as a form of unequal error protected, algebraic coding since it provided more coding protection to the set partitions with the smallest Euclidean distance.

Pottie and Taylor [31] collated and extended the above research of [27, 79–82] in a fundamentally important paper on multilevel coding. Here, the general construction of multilevel encoders and decoders is succinctly outlined. They point out that in a multilevel code, any code may be used in the

hierarchy of component codes. Code types may also be mixed – for example, block codes, convolutional codes and coset codes (including codes such as Ungerboeck codes) to name a few. The importance of signal set labelling is also shown. Finally, some of the trade-offs in terms of spectral efficiency, complexity and signal set expansion are presented. The work of [31] is perhaps the most useful base reference in this field to date. In parallel to and independently from [31], Calderbank [32] also produced a similarly important paper discussing multilevel code construction, decoding and related performance considerations.

Up to this stage, the literature is of a more general nature and sets in place the general multilevel coding theory. From here, there are two distinct branches of literature – those concerned with hierarchies of block component codes; and those investigating hierarchies of convolutional component codes. We concentrate on reviewing the myriad of papers relevant to multilevel block coding techniques.

Taylor and So [33] were at the forefront of a multitude of papers discussing specific multilevel code designs. They showed the design and performance of simple multilevel codes using a convolutional code combined with a zero-sum, block or convolutional self-orthogonal code as component codes for QAM constellations. The performance evaluation was on the AWGN channel. This paper demonstrated that the simulated performance agreed reasonably well with the calculated nominal coding gain in each case.

Kasami *et al.* [83] discussed the construction of multilevel block codes using 8-PSK and 16-QAM modulation schemes combined with Reed-Muller

component codes. They expanded the work of [31] by deriving lower bounds on the minimum distance of multilevel block codes.

Takata *et al.* [84] investigated various types of multistage decoding for multilevel block codes using M-ary PSK modulations. This paper also discusses the trade-off between spectral efficiency, error performance and decoding complexity. It concludes that multilevel decoding offers favourable trade-offs in terms of these parameters and that the codes discussed may be easily modified for trellis modulation codes.

Zhang and Vucetic have published several papers in this area, two of the more relevant ones, [85,86], are cited here. They describe a class of bandwidth efficient multilevel block codes for the Rayleigh fading channel. These codes are based on variable block-length Reed-Muller codes and are mapped onto quaternary-PSK (QPSK) and M-ary PSK ($M > 4$) constellations. They show that these codes outperform TCM codes with comparable decoder complexity and bandwidth efficiency. Gu *et al.* [87] continued this line of investigation, discussing multilevel codes using 16-QAM on the Rayleigh fading channel. This paper discusses methods of reducing performance degradation due to error propagation in the high level codes.

van Nobelen and Taylor [88] extended the concept of geometrically uniform codes to form geometrically uniform signal constellation partitions. They employ a combination of multiple symbol differential detection with multilevel coding to obtain a good error performance while attaining a channel state estimate. The channel used in this instance is the Rayleigh-Fading Channel.

Herzberg *et al.* [89] present and analyse concatenated multilevel coded

modulation schemes. At each component code level there are two codes – for example, an inner Reed-Solomon code and an outer binary block code. The constellation that is considered in this paper is 64-QAM. The drawback of this construction is that the component code block lengths are often very large and, therefore, decoding delay is relatively long. However, the paper shows that with careful system design, significant performance gains are achievable.

Wu and Costello [90] apply set partitioning to multidimensional constellations over GF(q) and construct multilevel codes using both block and trellis codes. Examples using Reed-Solomon component codes are cited. They discuss advantages of multistage decoding in terms of the available decoding options of each code – for example, soft- or hard-decision decoding at each level. It is also shown in this paper that if all component block codes are linear, the multilevel block code is also linear.

Isaka *et al.* [91] discuss multilevel coding for unequal error protection. This scheme is analysed using 16-QAM. It is intended for HDTV applications where some bits require more error protection than others.

Wachsmann *et al.* [92] have published the most recent paper discussing the broad area of multilevel coding concepts and design rules. This paper is also an excellent review of the subject area and includes the latest general multilevel coding methods with practical applications. Multilevel codes where one or more of the component codes are binary turbo codes are also discussed here and are shown to approach capacity at high bandwidth efficiencies. The trade-off between hard- and soft-decision decoding of component codes is also discussed.

The multilevel block coding literature to date has concentrated on the technical details of encoding and decoding multilevel codes. There have also been many papers, some of which have been highlighted, discussing the specific details of various component code hierarchies and decoding algorithms.

2.9.2 Combined Equalisation and Decoding

In this Thesis, we focus on the DFE for equalisation. In order to achieve improved performance on channels where equalisation is required, we consider schemes that assist in alleviating the error propagation within the feedback filter of the DFE, which limits the achievable coding gain. Two of the approaches for combating this problem are by using channel pre-coding at the transmitter or by using interleaving which introduces delay so that delayed reliable symbols may be used in the feedback filter [40, 93]. Transmitter pre-coding, while successful in some instances, is complicated for time varying channels (because adaptation is required) or even impossible for fast time varying channels where channel information is not known *a priori* [40, 41, 94]. The interleaving approach is not recommended for time critical applications as it suffers from delay [40]. Since the present Thesis is considering low complexity receiver structures for the slowly-varying, fixed-access, line-of-sight microwave radio channel, we consider combined equalisation and decoding at the receiver. The available literature on combined equalisation and decoding appears to be limited to a few key papers. These papers typically incorporate convolutional codes and a DFE combined with a trellis-type decoder.

Duel-Hallen and Heegard [95]; and Eyuboglu and Qureshi [70] inde-

pendently describe the use of delayed decision-feedback sequence estimation. These structures perform reduced-state Viterbi algorithms – often referred to as RSSE – for equalisation of linear ISI channels. This can also be extended to the coded case [96]. Chevillat and Eleftheriou [97] also discuss combined equalisation and TCM decoding in the form of a Viterbi decoder which operates on the combined ISI and code trellis. Again, independently of [70, 95], an efficient reduced-state decoding structure is described.

Yellin *et al.* [41] present a combined equalisation and decoding scheme using a DFE with one forward filter and multiple feedback filters. This structure has been described as a "tree" DFE. Here, the one path approximation to the Viterbi Algorithm is tracked and several "branches" and "sprouts" diverge from this path in an attempt to capture channel errors using one of the several feedback filters. This structure has significantly higher hardware complexity than a conventional DFE.

Gertsman and Lodge [98] use an iterative approach to joint demodulation and decoding. The system works on the principle that the demodulation and earlier decoding stages are able to refine their output with knowledge from the later stages by using an iterative approach to decoding. The code used in this system is a convolutional code that is decoded using the MAP symbol-by-symbol decoding algorithm. In this system, as the number of iterations increase the overall performance increases – with the biggest gain in performance being obtained between the first and second iterations.

Ariyavisitakul and Li [40] suggest joint convolutional coding and decision feedback equalisation based on the use of soft decisions and delayed

tentative decisions fed back into the DFE. This structure uses one forward filter and two feedback filters in the hybrid type DFE. The error correction decoder uses a standard Viterbi Algorithm. Reproduced in Figure 2.14 is the equalisation and decoding system used in [40]. It may be seen that one feedback filter uses instantaneous soft decisions obtained by passing the unquantised symbol estimate made by the DFE through a soft quantiser. These are used to cancel ISI and to adapt the DFE tap weights, however they do not cancel ISI perfectly since they are soft (unquantised) decisions. The second feedback filter uses delayed hard decisions from the output of the decoder. As they are hard and final decisions from the Viterbi decoder they are able to cancel ISI with more certainty than the previous soft decisions.

Recently there have been a number of papers, [99–101], investigating combined equalisation and decoding. These papers also use an iterative decoding process. They are based on the iterative turbo decoder and include de-interleaving. In [100] the data is convolutionally encoded and interleaved before being transmitted. The soft output from the convolutional decoder is utilised to reduce ISI and improve performance as the number of iterations increase. In the case of [101], there are no decoded symbols fed back into the equaliser as in the structure suggested in this Thesis.

All the papers described here yield performance improvements over conventional equalisation and decoding. As is apparent from this literature survey, there is limited literature on combining equalisation and decoding of block codes. Furthermore, until recently there were no papers incorporating an iterative approach to joint demodulation and decoding. Recently, itera-

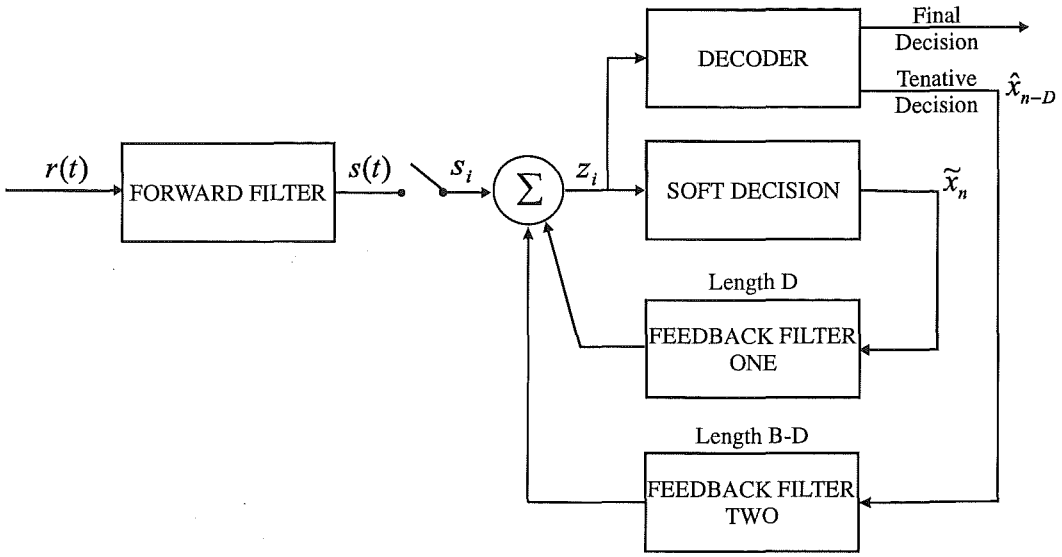


Figure 2.14: Joint coding and DFE receiver model proposed in [40]. B is the total number of feedback filter taps.

tive combined decoding and detection work has appeared in the code-division multiple access (CDMA) literature [102]. The overall topic is of interest since the combination of equalisation and decoding may help overcome some of the performance limitations when performing these functions independently.

2.10 Summary

This Chapter has provided a summary of the components in a digital communications system (cf. Figure 2.1) and the general coding concepts used

throughout this Thesis as follows:

- Quadrature amplitude modulation (QAM), a commonly used constellation used in digital communications.
- Efficient transmitter and receiver pulse shaping.
- Equalisation techniques, in particular decision-feedback equalisation (DFE), for the mitigation of intersymbol interference (ISI) introduced by a multipath channel.
- Error correcting coding principles with the focus on linear block codes and more specifically Reed-Muller block codes, which are used in the multilevel codes examples presented in this Thesis.
- The concepts of coded modulation, which is a method of bandwidth conservation while saving power.
- Ungerboeck set-partitioning and multidimensional constellations as applied to coded modulation.

Finally, in this Chapter we have provided a brief literature review of the papers relevant to this Thesis in the subject areas of multilevel coding and joint equalisation and decoding.

Chapter 3

MULTILEVEL CODING

3.1 Introduction

Here we consider high-rate coded signalling formats employing spectrally efficient modulation schemes [31, 44]. Their intended use is in high-integrity, fixed-access wireless systems transmitting via line-of-sight microwave radio channels. Usually these systems have strict bandwidth and power limitations, use low gain antennas and often require significant coding gain with a short decoding delay to achieve the necessary performance levels. This Chapter develops coded modulation schemes that utilise simple encoding and decoding structures.

Following the work of [27,31–33], we apply multilevel block coding techniques to the incoming data stream using a set of component codes mapped to an expanded QAM signal set. In the present work, the component codes used are Reed-Muller block codes – however, other sets of block codes could equally well be used. As will be seen, multilevel block codes can offer significant coding

gain and minimal delay with modest decoding complexity [31, 32].

3.2 Multilevel Codes

Multilevel coding is a technique for combining coding and modulation. It was initially proposed by Imai and Hirakawa in 1977 [27] and the techniques have been further refined in works such as [31–35]. Multilevel codes may be constructed using an arbitrary hierarchy of component codes mapped to an expanded signal set [31, 32]. By distributing coding complexity over several simple component codes, overall code complexity is reduced [27, 31–33, 44, 77]. Multilevel codes provide an alternative to TCM [28] in the construction of bandwidth efficient coded signal sets.

The hierarchy of component codes in a multilevel code may be denoted $\mathbf{C} \equiv (C_1, C_2, \cdots, C_m)$, where m is the total number of codes. These component codes may be of any type – such as convolutional codes, block codes or concatenated codes. As with TCM, multilevel codes also allow for parallel transitions consisting of uncoded bits. Different coding schemes may be used in one multilevel code [31]. Multilevel codes typically map their component codes to signal set partitions. The signal constellation employed may be any modulation scheme that may be set partitioned – for example, QAM, PSK or pulse amplitude modulation (PAM).

In this context we partition the original constellation, S_0, $(m-1)$ times to obtain the subset chain $S_0/S_1/\cdots/S_{m-1}$ [28, 44] such that

$$\delta_0 \leq \delta_1 \leq \cdots \leq \delta_{m-1} \tag{3.1}$$

where δ_l^2 is the MSED between elements in the subset of S_l at partition level l. Ideally, each point associated with each subset level will have increasing Euclidean distance such that

$$\delta_0 < \delta_1 < \cdots < \delta_{m-1} \tag{3.2}$$

As will be seen, given the correct choice of component codes this allows us to design codes to achieve equal performance at each partition level. That is, as the partition level increases, a less powerful code (with a higher code rate and smaller Hamming distance) may be used and equal performance maintained at each level.

In a multilevel code, each component block code may be compactly represented as $C_\nu = (n_\nu, k_\nu, d_{min,\nu})$, where n_ν is the encoded block length of the component code; k_ν is the number of information bits; and $d_{min,\nu}$ is its minimum Hamming distance. Using the staged encoder structure of [31] (cf. Figure 3.1), each component code, C_ν, encodes k_ν information bits to n_ν coded bits to form a portion of the composite code word. For the AWGN channel, an efficiently coded multilevel coded structure will have the minimum Hamming distance of each code inversely proportional to the MSED between signal points at each partition level. In other words, the first code, C_1, will have the largest minimum Hamming distance; the second code, C_2 will have the next largest Hamming distance; and so on through the codes [86]. Each word of C_ν chooses a sequence of subsets associated with the constellation partition $S_{\nu-1}/S_\nu$. Each code may be decoded using either hard- or soft-decision decoding depending on performance and complexity requirements.

We focus on the use of block codes. Block codes have features that may

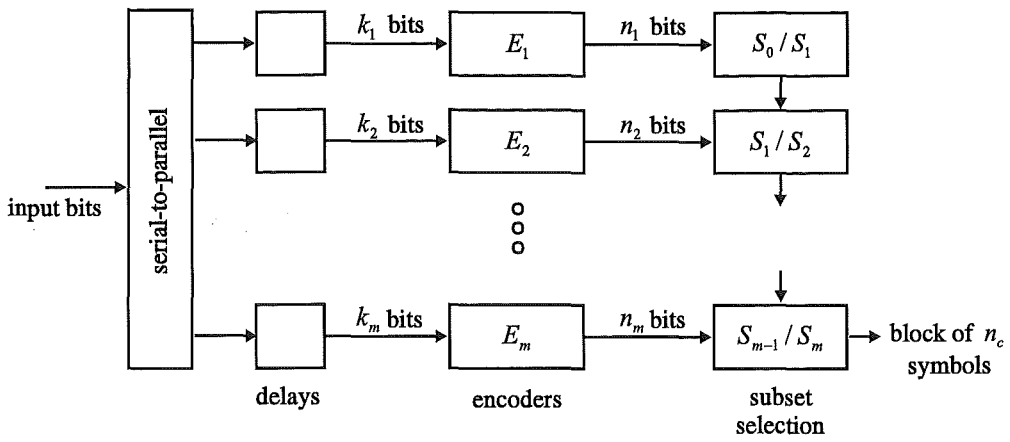


Figure 3.1: Block diagram of staged encoder structure for a multilevel code.

be utilised advantageously in a multilevel block coding structure. For example, a block code will have a periodically time varying trellis. At the conclusion of decoding each block of symbols, the system is back to its original state and has a constant decoding delay time for each block, furthermore, the remaining path through the trellis is the maximum likelihood path. This is convenient in the case of framed transmission as found in time-division multiple access (TDMA). On the other hand, convolutional or trellis codes may take a variable time to decode or require trellis truncation. In the case of framed data being transmitted, the decoder may have to be flushed with tailing symbols at the end of each frame, thus increasing system overhead.

In the present work, we choose to utilise Reed-Muller block codes as component codes. Given our design objectives of low complexity and short decoding delay, we propose the use of low order Reed-Muller codes with a short block length [23]. These codes are simple to decode and help restrict decoding delay. One of the problems associated with many Reed-Muller codes is the possibility of codes having long runs of zeros and ones, resulting in long repetitive channel sequences with poor spectral properties. As will become clear, the multilevel mapping process assigns each component code to different signal set partition levels so this problem is alleviated.

We arrange the encoded symbols into a block size equal to the block length of the largest component code. To simplify the mapping of bits to symbols, we generally choose all component codes to have a common block length, n_c. The common block length is chosen such that the number of two-dimensional symbols transmitted in each block, n_c/N, is an integer. However, it is possible to have component codes with any block length.

Our objectives when constructing a multilevel block code are to achieve high performance, low decoder complexity, minimal code overhead and a short decoding delay time. We also aim to achieve essentially the same spectral efficiency as TCM systems of similar throughput. The MSED of a multilevel block code, $D^2_{min}(\mathbf{C})$, determines the system's performance and is given by [33]

$$D^2_{min}(\mathbf{C}) = \min_{\nu}\{d_{min,1}\delta^2_0, \cdots, d_{min,m}\delta^2_{m-1}\} \tag{3.3}$$

where $d_{min,\nu}$ is the minimum Hamming distance of code C_ν. The error correcting and detecting capability of each code is dependent on $d_{min,\nu}$ for hard-decision decoding and on $d_{min,\nu}\delta^2_{\nu-1}$ for soft-decision decoding. The MSED of

the coded partition level $S_{\nu-1}/S_\nu$ is

$$D^2_{min}(C_\nu) = d_{min,\nu}\delta^2_{\nu-1} \tag{3.4}$$

The nominal coding gain, γ_0, of a multilevel block code is found from Equation (3.3). It is achieved at asymptotically high SNR, assuming soft-decision decoding, and is defined as [31,33]

$$\gamma_0 = 10\log_{10}\left(\frac{D^2_{min}(\mathbf{C})}{d^2_{ref}}\right) \tag{3.5}$$

where d^2_{ref} is the MSED between the signal points of an uncoded reference signal set with the same average power.

The rate of each component code, C_ν, as given in Equation (2.32), is

$$R_\nu = \frac{k_\nu}{n_c} \tag{3.6}$$

From Equation (3.6) we may calculate the effective code rate of a multilevel block code, $R_{\mathbf{C}}$. This is achieved by considering the ratio of the total number of information bits encoded to the total number of coded bits. This yields

$$R_{\mathbf{C}} = \frac{\sum_{\nu=1}^{m}\frac{k_\nu b_\nu}{n_c}}{\sum_{\nu=1}^{m} b_\nu} \tag{3.7}$$

where b_ν is the number of encoded bits per two-dimensional symbol in each partition $S_{\nu-1}/S_\nu$; k_ν is the number of information bits in each component code word; n_c is the common block length of the component codes; and m is the number of partition levels.

3.3 Encoding a Multilevel Block Code

Encoding of a multilevel block code is by means of a staged encoder structure as shown in Figure 3.1 [31]. We construct two multidimensional multilevel block codes as examples. The first is based on a four-dimensional 96-QAM constellation and the second on an eight-dimensional 80-QAM constellation. The two differ in their achievable throughput and in the dimensionality over which shaping is performed.

We consider first a four-dimensional multilevel block code to transmit $\rho = 5$ bits/symbol[1] based on a 96-QAM constituent constellation consisting of $2^6 = 64$ inner points and $2^6/2 = 32$ outer points as in Figure 2.13(a). The 96-QAM constellation is partitioned into the chain of subsets $S_0/S_1/S_2/S_3$ and employs three Reed-Muller component codes, namely $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$ and $C_3 = (32, 26, 4)$, where $n_c = n_\nu = 32$ so that each coded signal block contains $n_c/N = 16$ four-dimensional points. From Equation (3.3) the codes C_1, C_2 and C_3 each achieve an inter-subset MSED of the code of $D^2_{min}(C_\nu) = 16\delta_0^2$. However, the intra-subset distance at partition level S_3 is $D^2_{min}(C_4) = 8\delta_0^2$. Since we do not apply a code to the subsets at this level - that is, C_4 is equivalent to a rate one code - it restricts the overall MSED of the code to $D^2_{min}(\mathbf{C}) = 8\delta_0^2$ as implied in Equation (3.3)[2].

The code, $C_1 = (32, 6, 16)$, is a first order Reed-Muller code that encodes $k_1 = 6$ data bits into $n_1 = 32$ encoded bits identifying a block of 16

[1] In this Thesis, throughput is always expressed in terms of the basic two-dimensional constellation.

[2] As will be shown in Chapter 4, it is often better to let a smaller $D^2_{min}(\mathbf{C})$ dominate the exponential behaviour of the code performance in order to obtain a reduced error coefficient and overall improved code performance [34].

four-dimensional subsets at partition level S_0/S_1. In other words, C_1 chooses the LSB of the 32 two-dimensional symbols in the signal block. Since we intend here to use a four-dimensional shaping code, we choose to employ a four-dimensional labelling format represented by two two-dimensional symbols. Therefore, the label for the partially encoded four-dimensional symbol is represented as $(- - - - - - - \; x_{1,j})$, $(- - - - - - - \; x_{1,j+1})$, where $x_{1,j}$ and $x_{1,j+1}$ denote the consecutive LSB's chosen by C_1, and $-$ denotes the bits in each symbol not yet labelled.

The second code, $C_2 = (32, 16, 8)$, is a second order Reed-Muller code encoding $k_2 = 16$ data bits into $n_2 = 32$ encoded bits. Given the subsets selected by C_1, 16 four-dimensional subsets are selected from signal set partition S_1/S_2. The partial four-dimensional symbol label now has the form $(- - - - - \; x_{2,j} \; x_{1,j})$, $(- - - - - \; x_{2,j+1} \; x_{1,j+1})$, where $x_{2,j} \; x_{1,j}$ and $x_{2,j+1} \; x_{1,j+1}$ denote consecutive bits chosen by C_2 and C_1.

The third code, $C_3 = (32, 26, 4)$, is a third order Reed-Muller code encoding $k_3 = 26$ data bits into $n_3 = 32$ encoded bits. Again, 16 four-dimensional subsets are selected from partition S_2/S_3 given the subset selection of C_2 and yields the partial four-dimensional symbol label in the form $(- - - - \; x_{3,j} \; x_{2,j} \; x_{1,j})$, $(- - - - \; x_{3,j+1} \; x_{2,j+1} \; x_{1,j+1})$ where $x_{3,j} \; x_{2,j} \; x_{1,j}$ and $x_{3,j+1} \; x_{2,j+1} \; x_{1,j+1}$ denote the consecutive bits chosen by C_3, C_2 and C_1.

The first three codes encode a total of 48 data bits. To this point we have achieved a throughput of $\rho = 1.5$ bits/symbol, which could be transmitted using 32 symbols of an 8-point two-dimensional signal constellation. To achieve a higher throughput, parallel transitions can be arbitrarily mapped to each

block of 32 signal points. To reduce the transmitted PAR let us view the block of 32 symbols as 16 four-dimensional symbols by treating pairs of points from S_3 as four-dimensional signal points [44]. To achieve a throughput of $\rho = 5$ bits/symbol we use a rate one code, $k_4 = n_4 = 32$, to map the remaining 3.5 bits/symbol to the block of 16 four-dimensional symbols. In effect, this is the addition of parallel transitions to the code which maps the $3.5n_4 = 112$ bits over the four-dimensional constellation in 7-bit blocks, $(l_1, l_2, \cdots, l_7)$, as follows:

1. If $l_1 = 0$, two two-dimensional inner points are used to construct the four-dimensional symbol. The MSB of each two-dimensional symbol is set to zero. The last six bits, $(l_2, \cdots, l_7)$, select the remaining three bits in each symbol. The overall label is then given by $(0\, l_2\, l_3\, l_4\, x_{3,j}\, x_{2,j}\, x_{1,j})$, $(0\, l_5\, l_6\, l_7\, x_{3,j+1}\, x_{2,j+1}\, x_{1,j+1})$.

2. If $l_1 = 1$, the four-dimensional symbol consists of a two-dimensional inner point and a two-dimensional outer point. If also $l_2 = 1$, the first two-dimensional symbol is chosen as the outer point, its two high order bits are set to one and the MSB of the inner point is set to zero. (l_3, l_4) select the remaining two bits of the outer point and (l_5, l_6, l_7) select the final bits of the inner point. This gives the complete four-dimensional label $(11\, l_3\, l_4\, x_{3,j}\, x_{2,j}\, x_{1,j})$, $(0\, l_5\, l_6\, l_7\, x_{3,j+1}\, x_{2,j+1}\, x_{1,j+1})$.

3. If $l_1 = 1$ and $l_2 = 0$, the second two-dimensional symbol is selected as the outer point. The remaining bits are allocated in a similar way to step 2, selecting the remaining bits of the four-dimensional symbol to

give the label $(0\, l_3\, l_4\, l_5\, x_{3,j}\, x_{2,j}\, x_{1,j})$, $(11\, l_6\, l_7\, x_{3,j+1}\, x_{2,j+1}\, x_{1,j+1})$.

The shaping is added as parallel transitions at partition level S_3 where points are separated by only the intra-subset distance $8\delta_0^2$. Thus, the distance of the overall code is expressed as $D^2_{min}(\mathbf{C}) = 8\delta_0^2$ rather than the inter-subset distance of $D^2_{min}(C_\nu) = 16\delta_0^2$ achieved by C_1, C_2 and C_3.

In principle, a simple method of increasing or decreasing the throughput of a multilevel block code is to alter the constellation size and/or dimensionality while using a similar construction of component codes. For example, we may use 80-QAM ($2^6 = 64$ inner points and $2^6/4 = 16$ outer points – cf. Figure 2.13(b)) with an eight-dimensional shaping code. Using the same low level component codes as in the previous 96-QAM case, this combination achieves a throughput of only $\rho = 4.75$ bits/symbol: however, its constituent two-dimensional constellation has a smaller PAR ($\text{PAR}_{80} = 2.85$ dB) compared to 96-QAM ($\text{PAR}_{96} = 3.22$ dB) and, as will be seen in Chapter 4, a slightly better bit error performance as a function of SNR (E_b/N_0).

In the second example, the 80-QAM multilevel block code maps a block of 152 information bits to eight eight-dimensional symbols. The allocation of bits for the low level codes is the same as for 96-QAM. However, to achieve the desired throughput we use a rate one code, $k_4 = n_4 = 32$, to map the remaining 3.25 bits/symbol over the block – that is, we have $3.25n_4 = 104$ bits in total. These bits are divided into eight 13-bit blocks denoted $(l_1, \cdots, l_{13})$ mapping the remaining 104 bits using an eight-dimensional shaping algorithm as follows:

1. If $l_1 = 0$, four two-dimensional inner points are used in the eight-

dimensional symbol. The MSB of each two-dimensional symbol is set to zero. The last twelve bits $(l_2, \cdots, l_{13})$ select the remaining three bits in each symbol giving $(0\, l_2\, l_3\, l_4\, x_{3,j}\, x_{2,j}\, x_{1,j})$, $(0\, l_5\, l_6\, l_7\, x_{3,j+1}\, x_{2,j+1}\, x_{1,j+1})$, $(0\, l_8\, l_9\, l_{10}\, x_{3,j+2}\, x_{2,j+2}\, x_{1,j+2})$, $(0\, l_{11}\, l_{12}\, l_{13}\, x_{3,j+3}\, x_{2,j+3}\, x_{1,j+3})$ as the overall label. Since we are using an eight-dimensional shaping code, we have employed an eight-dimensional labelling format represented by four two-dimensional symbols.

2. If $l_1 = 1$, the eight-dimensional symbol consists of three two-dimensional inner points and one two-dimensional outer point. We use the next two bits in the sequence to determine the position of the outer point in the sequence of four two-dimensional symbols. For example, if $l_2 = 0$ and $l_3 = 0$, the first two-dimensional symbol is chosen as the outer point. The three high order bits of its label are set to one and the MSB of the inner point labels are set to zero. l_4 selects the remaining bit of the two-dimensional outer point label and $(l_5, \cdots, l_{13})$ select the final three bits of the three remaining (inner) point labels. The overall eight-dimensional label is then given by $(111\, l_4\, x_{3,j}\, x_{2,j}\, x_{1,j})$, $(0\, l_5\, l_6\, l_7\, x_{3,j+1}\, x_{2,j+1}\, x_{1,j+1})$, $(0\, l_8\, l_9\, l_{10}\, x_{3,j+2}\, x_{2,j+2}\, x_{1,j+2})$, $(0\, l_{11}\, l_{12}\, l_{13}\, x_{3,j+3}\, x_{2,j+3}\, x_{1,j+3})$. The bit allocations for the other possible combinations of (l_2, l_3) are similarly used to select the position of the outer point in the eight-dimensional symbol.

Finally, if we wish to increase the throughput to $\rho = 6$ bits/symbol, we may enlarge the 96-QAM constituent constellation to 192-QAM (128 inner and 64 outer points) with a four-dimensional shaping code. Similarly, to

increase throughput to $\rho = 5.75$ bits/symbol, we may enlarge the 80-QAM constituent constellation to 160-QAM (128 inner and 32 outer points) with an eight-dimensional shaping code. In these two situations, if the average two-dimensional symbol energy of S_0 remains unity, the MSED between adjacent signal points is reduced and there is a small performance loss due to constellation expansion.

Two-dimensional constellations may also used in conjunction with multilevel block codes. As an example we may encode 64-QAM employing the same three Reed-Muller component codes, $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$ and $C_3 = (32, 26, 4)$, as previously. Again, these codes encode the three LSB's of each symbol label as in the 80-QAM and 96-QAM examples. Since a shaping code costs throughput and uses a more complex constellation, we use a rate one encoder and no shaping code to encode the remaining bits as parallel transitions on the last three most significant bits.

3.4 Decoding a Multilevel Block Code

Following [27,31], a staged decoder structure is used to decode a multilevel block code, as shown in Figure 3.2. All decoders may in principle utilise hard- or soft-decision decoding. In each block of received symbols the first decoder, D_1, operates on the received signal to make a decision on the sequence of subsets in S_1. The second decoder, D_2, uses both the decisions from D_1 and the original received signal sequence to make a decision on the sequence of subsets in S_2. This process continues through all code levels until the entire block of symbols has been decoded.

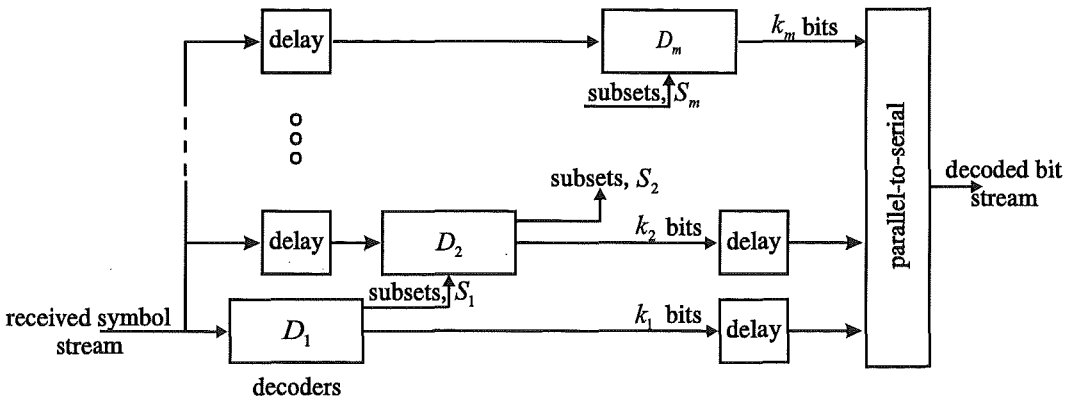


Figure 3.2: Block diagram of staged decoder structure for a multilevel code.

3.4.1 Decoding the First Level Block Code

Decoder D_1 decodes C_1 and may use hard-decision decoding, where the information bits are decoded using a predefined look-up table or the Reed decoding algorithm [23]. Since the set of possible code words in C_1 is often small we may easily hard-decision decode it using a look-up table. The Reed decoding algorithm is usually used to decode codes having many code words where the look-up table becomes too cumbersome.

C_1 may also be decoded using soft-decision decoding. In the present work, C_1 is usually soft-decision decoded since it is often a code with a large Hamming distance to compensate for the small Euclidean distance between signal points in the subsets at partition level S_0/S_1. A large Hamming distance implies that the code has correspondingly good error correcting proper-

ties. This is advantageous when soft-decision decoding is used. Soft-decision decoding uses the Euclidean distance of the received signals from the possible set of transmitted signals as a metric to determine the correct output. The metric uses only one (the nearest) signal point in the set of possible signals representing the corresponding bit. In order to investigate the trade-off in terms of performance versus complexity, we implement two different soft-decision decoding methods – an optimal and a sub-optimal algorithm as discussed in Section 2.7. Since we are encoding a sequence of subsets at each partition level we use Viterbi-based trellis decoding. This is the optimal soft-decision decoding for sequences in terms of performance [66] and provides, in this case, minimum word error probability.

The second soft-decision decoder we examine is a much simpler, but sub-optimal, algorithm based on the Wagner decoding technique [71]. The Wagner-based decoding algorithm utilises the property that Reed-Muller codes are quasi-perfect [13]. Quasi-perfect codes have the capability of correcting all error patterns of weight less than or equal to ω, where $\omega = \lfloor (d_{min,\nu} - 1)/2 \rfloor$. They are also able to detect all, and correct some, error patterns of weight $(\omega + 1)$ [45]. In addition, a significant number of error patterns of weight greater than $(\omega + 1)$ may be detected [13, 23]. Thus, by using the Wagner-based decoding algorithm, which utilises some reliability information in the decoder, we may obtain some performance gain by detecting and correcting some error patterns of weight greater than $(\omega + 1)$.

The Wagner-based decoding algorithm initially uses a look-up table to determine how many bits in the received sequence are in error. If there are ω or

fewer bits in error, they are corrected immediately via the look-up table. Given $\frac{d_{min,\nu}}{2}$ detected errors, the Wagner-based decoding algorithm is implemented. We have found by simulation that improved performance is obtained when implementing it more than once. Our simulation work has indicated that a good choice for the number of times the Wager-based decoding algorithm is implemented is Ω, where $\Omega = \lceil d_{min,\nu} - \frac{d_{min,\nu}}{4} - 1 \rceil$; and $d_{min,\nu} \geq 2$. It transpires that the more powerful a code is the better the Wagner-based algorithm will work, but it may cost more in terms of the number of iterations performed. Note that when $d_{min,\nu} = 4$, the Wagner-based algorithm may perform two iterations and when $d_{min,\nu} = 8$, the Wagner-based algorithm performs as many as five iterations.

In operation, the Wagner-based decoding algorithm complements the bit associated with the least reliable received symbol – that is, the symbol with the largest Euclidean distance between the received and allowed signal points. The new bit sequence is checked again via the look-up table to ascertain if the error count has been reduced. If the error count has been reduced but is still greater than ω, this process is repeated for the bit with the next least reliable symbol until ω or fewer bits remain in error. The remaining ω bits are corrected by using the look-up table. Should at any point the inverted bit not reduce the error count, it is restored and the bit with the next least reliable symbol is inverted instead. As applied here, the Wagner-based algorithm is restricted to the inversion of at most Ω bits so as to restrict decoder complexity and to retain a short decoding delay.

In practice, it has been observed that there is little resulting improve-

ment from iterating the process more often than Ω and this restricted usage recovers much of the soft-decision decoding advantage at high SNR. Should the Wagner-based decoding algorithm not reduce the error count to ω bits within Ω iterations, the code word closest to the original received bit sequence is selected and errors are likely to occur. We have found that the Wagner-based decoding algorithm can result in more than ω errors being corrected.

3.4.2 Decoding Higher Level Block Codes

Decoder D_2 decodes C_2 and may also utilise either hard- or soft-decision decoding. At this level, the code often contains a large number of code words, and therefore, usually hard-decision decoding is performed using the Reed decoding algorithm [23] rather than a look-up table. If one chooses to soft-decision decode C_2, Viterbi trellis decoding is implemented. Depending on the code used at this level, its trellis structure may be quite complex with a large number of states.

Decoder D_3 hard-decision decodes C_3, usually using the Reed decoding algorithm. Since this code usually has a very large number of code words, soft-decision trellis decoding would result in a very complex trellis structure as discussed in Section 2.7.2. Moreover, because of the increased subset distance, hard-decision decoding provides most of the available performance.

The trade-off between soft- and hard-decision decoding each higher level code must be explored. In general, the higher level codes are also higher rate codes with smaller Hamming distances than the first level code. On the other hand, the Euclidean distances between subsets at each level are larger than the

first level code. The detection of the correct received signal at the higher levels relies heavily on the increased Euclidean distance between signal points as well as on the Hamming distance of the codes. In general, most of the performance gain is due to the larger Euclidean distance at each higher level. Also, at these levels the code trellis structures are significantly more complex to decode since they generally have a large number of code words which results in a trellis structure with many possible states and paths. In these cases, hard-decision decoding is usually relatively simple to implement when using algorithms such as the Reed decoding algorithm – therefore, we usually hard-decision decode the higher level codes.

Finally, if we are implementing a multidimensional constellation, we decode the shaping code into a sequence of N-dimensional points. The decoder identifies the received data sequence by decoding the inner and outer point layout of the shaping code. It is effectively the reverse of the mapping process given in Section 3.3. If a two-dimensional constellation is being utilised, the remaining parallel transitions are simply demodulated and hard-decision detected.

3.5 Summary

This Chapter has described the construction and encoding techniques of multilevel codes. In particular we have focused on multilevel block codes utilising Reed-Muller component codes and large QAM constellations. As is shown, there is no restriction on the size (and dimensionality) of the QAM constellations as long as they can be successfully set partitioned as per the

requirements of multilevel coding. Finally, we have discussed various decoding techniques relevant for component codes in a multilevel block code.

Chapter 4

PERFORMANCE EVALUATION: AWGN CHANNEL

4.1 Introduction

In this Chapter we evaluate the performance of multilevel block coded modulation employing QAM constellations and Reed-Muller component codes. We consider the performance-complexity trade-off between soft-decision trellis decoding, Wagner-based decoding and hard-decision decoding. The channel considered here is the AWGN channel. An analytical approximation to the error performance of multilevel block codes on an AWGN channel is described and compared to simulation results. Using several large QAM constellations we show that the analytical approximation gives a good performance estimate at moderate to high SNR. The effect of parallel transitions is considered and we explore the trade-off between distance and the error coefficient.

The performance curves in this Chapter show the overall BER as a

function of SNR. In each case, the SNR is in terms of average bit-energy-to-noise ratio (cf. Section 2.4.3) and is again reproduced here for convenience[1]

$$\begin{aligned} \text{SNR} &= 10\log_{10}\left(\frac{E_b}{N_0}\right) \\ &= 10\log_{10}\left(\frac{1}{R\log_2 M} * \frac{E_{sym}}{N_0}\right) \end{aligned} \tag{4.1}$$

where M is the number of signal points in the constituent constellation; E_b is the average bit energy; N_0 is the power spectral density of the noise; R is the code rate; and E_{sym} is the average transmitted two-dimensional symbol energy of the coded constellation. Due to the shaping code, E_{sym} is slightly less than the average symbol energy for the constituent two-dimensional constellation.

4.2 Performance of 96-QAM using Alternative Decoding Schemes

Here we evaluate via simulation the performance of the four-dimensional 96-QAM multilevel block code case described in Section 3.3 – which achieves a throughput of $\rho = 5$ bits/symbol. The Reed-Muller component codes employed are $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$ and $C_3 = (32, 26, 4)$. A four-dimensional shaping code is also used. We consider three decoding variations for the first decoder – namely, hard-decision decoding; Wagner-based decoding; and soft-decision trellis decoding. The uncoded reference constellation used in these cases is uncoded 32-QAM which has a throughput of $\rho = 5$ bits/symbol.

[1]SNR usually refers to $\frac{E_s}{N_0}$, where E_s is the average symbol energy. However here SNR is given in terms of average bit-energy-to-noise ratio, $\frac{E_b}{N_0}$, which enables a fair comparison of the power efficiency of different schemes – for example, coded and uncoded systems with similar throughput.

From Figure 4.1 at a BER of 10^{-5}, the coding gains for hard-decision decoding C_1 and trellis decoding C_1 is approximately 2 dB and 2.75 dB respectively over uncoded 32-QAM. Decoding C_1 using the Wagner-based decoding algorithm yields performance that is slightly better than hard-decision decoding and appears to be converging at high SNR to that of trellis decoding. At high SNR, the Wagner-based decoding algorithm primarily uses a look-up table and then occasionally implements the Wagner process if it is required. Therefore, it may yield significant savings in terms of decoding computation complexity and delay compared to trellis decoding. For low order Reed-Muller codes, the Wagner-based decoding algorithm is simple to implement, relatively inexpensive computationally and on average has a short decoding delay time. It may be the preferred decoding algorithm at high SNR from a speed and complexity standpoint. However, soft-decision trellis decoding is usually the preferred option to achieve a consistent coding gain at low SNR and a constant and predicable decoding delay over all SNR.

Figure 4.2 illustrates that trellis decoding both C_1 and C_2 yields an additional coding gain of about 0.5 dB at a BER of 10^{-5}. When soft-decision decoding is implemented on standard block or convolutional codes there is approximately a 2-dB advantage over hard-decision decoding at high SNR [13]. Since we are using a large constellation and are soft-decision decoding only a portion of the constituent two-dimensional symbol, we would not expect to obtain the overall 2-dB gain. Whether or not to soft-decision decode each code is a design trade-off between decoder complexity and performance. Soft-decision decoding the higher level codes tends to provide diminishing returns

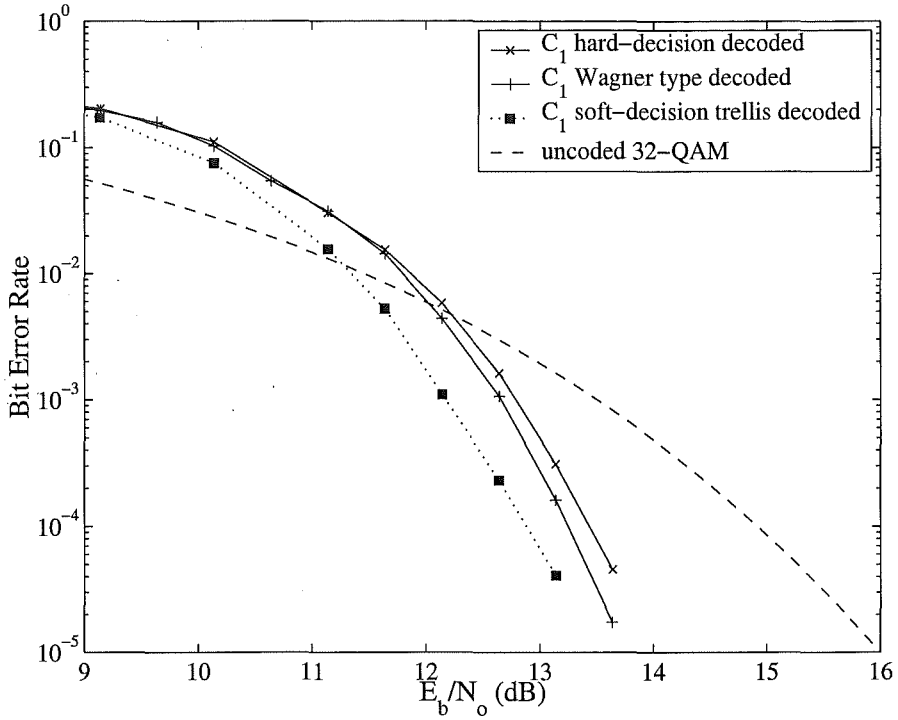


Figure 4.1: Simulated performance of a 96-QAM multilevel block code with variations in methods for decoding C_1. All remaining high level codes are hard-decision decoded.

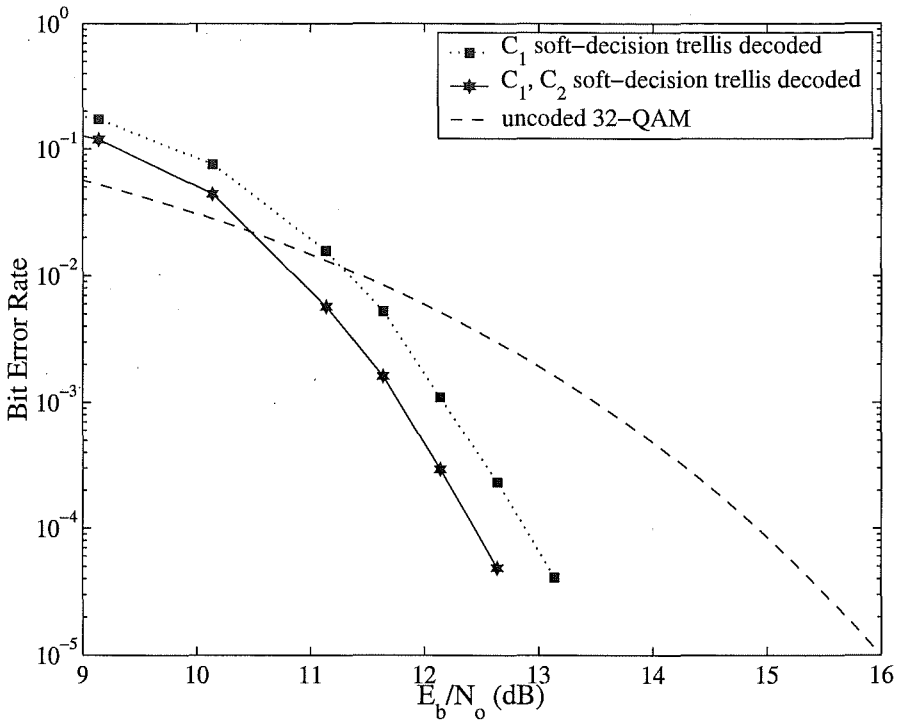


Figure 4.2: Simulated performance of a 96-QAM multilevel block code, soft-decision trellis decoding C_1 and C_2. All remaining high level codes are hard-decision decoded.

in terms of performance versus complexity. At the higher partition levels, the Euclidean distance between the subsets is large – which provides a lot of the performance margin. Also, the soft-decision trellis decoder of the higher level codes will have high computational complexity as discussed in Sections 2.7.2 and 3.4.2.

4.3 Approximation to Error Performance

We develop here an analytical approximation to the error performance of multilevel block codes. The error performance, P_e, of any multilevel code may be bounded as [31]

$$P_e \leq \max_{\nu}\{P_e(C_1), P_e(C_2), \cdots, P_e(C_m)\} \tag{4.2}$$

in terms of component code performance. The error performance of each component code may, at moderate to high SNR, be approximated by [13]

$$P_e(C_\nu) = N_{d_{min,\nu}}\, Q\left(\sqrt{\frac{D^2_{min}(C_\nu)}{2N_0}}\right) \tag{4.3}$$

where $Q(\bullet)$ is the Gaussian error probability (Q-function); $N_{d_{min,\nu}}$ is the error coefficient of the code; N_0 the noise power spectral density; and $D^2_{min}(C_\nu) = d_{min,\nu}\delta^2_{v-1}$ is the MSED of the component code given in Equation (3.4). Since the Q-function dominates the outcome of Equation (4.2), the performance of the overall multilevel code may be approximated by substituting Equation (4.3) into Equation (4.2) to give

$$P_e \cong N_{d_{min,\nu}}\, Q\left(\sqrt{\frac{D^2_{min}(\mathbf{C})}{2N_0}}\right) \tag{4.4}$$

where $D^2_{min}(\mathbf{C})$ is the MSED of the code given in Equation (3.3); and $N_{d_{min},\nu}$ is the error coefficient of the last non-trivial code which determines the path structure, and therefore the number of nearest neighbours, between parallel transitions.

For example, when using a multilevel block code utilising large QAM constellations and component codes $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$ and $C_3 = (32, 26, 4)$, the parallel transitions (which may be considered as a trivial code, C_4) at distance $8\delta_0^2$ (subset distance of S_3) determine the exponential behaviour. At partition level three, the subset sequence in each block is determined by C_3, and therefore the error coefficient in Equation (4.3) is that of C_3. In these examples, further partitioning to increase subset distance is not possible. Thus, overall error performance is bounded by $P_e(C_4)$, which at moderate SNR, is given by Equation (4.3). In principle, some improvement can be obtained by using a non-trivial code for C_4 – such as a $(32, 31, 2)$ single parity check code or another code with a block length $n_c = 32$. However, this results in a loss of throughput and increases the error coefficient of the overall code [33]. If we use codes C_3 (or C_4) which have large error coefficients, the resulting error coefficient of the overall code will also be very large. This can lead to a performance loss, particularly at low to moderate SNR. In many cases it is better to use lower level codes with more distance than required and not to use a code C_4 in order to obtain a moderate error coefficient. Alternatively, we may use C_4 for constellation shaping, as noted previously, rather than as an attempt to gain distance.

For the actual multilevel block codes being considered in this Thesis, C_3

is a third order Reed-Muller code. Its error coefficient may be calculated from the general formula for the error coefficient of an r^{th} order Reed-Muller code of length 2^m – cf. Equation (2.43) [30, 45]. For $C_3 = (32, 26, 4)$, this results in $A_{d_{min},3} = 1240$. We may then approximate the error performance of the overall code by replacing $N_{d_{min},\nu}$ in Equation (4.4) with $A_{d_{min},3}$ and substituting for $D^2_{min}(\mathbf{C})$ using knowledge of the average energy of the signal set to obtain the approximation [34, 35]

$$P_e \cong A_{d_{min},3}\, Q\left(\sqrt{\frac{R_{\mathbf{C}} \log_2 M}{2x} \frac{E_b}{N_0}}\right) \tag{4.5}$$

where x is the ratio of the average energy per two-dimensional signal constellation to the MSED of the code, $D^2_{min}(\mathbf{C})$. Note that Equation (4.5) presupposes correct decoding by all lower level decoders in the staged decoder.

4.4 Analytical versus Simulated Performance

We now compare multilevel block code performance, estimated using Equation (4.5), to simulated performance. In the following examples we use a range of constellation sizes and dimensions as well as several component code combinations. In each example we use one of the hierarchies of three Reed-Muller component codes defined in Table 4.1.

In these examples, C_1 is always soft-decision decoded using a trellis decoder and C_2, C_3 are hard-decision decoded using the Reed decoding algorithm. The parallel transitions are decoded based on the shaping code for the multidimensional constellations. For the two-dimensional constellation cases we do not use a shaping code, and the parallel transitions are simply detected

Table 4.1: Three component code hierarchies, also shown is the error coefficient for C_3.

code hierarchy	C_1	C_2	C_3	$A_{d_{min},3}$
case one	(32,6,16)	(32,16,8)	(32,26,4)	1240
case two	(16,5,8)	(16,11,4)	(16,15,2)	2^{15}
case three	(16,1,16)	(16,5,8)	(16,11,4)	140

as hard decisions.

For each coded case we also compare performance estimated using the analytical approximation of Equation (4.5) to simulated error performance. The error coefficient of the last non-trivial code, $A_{d_{min},3}$, in each of these cases is shown in Table 4.1. The overall MSED of each multilevel code is $D^2_{min}(\mathbf{C}) = 8\delta_0^2$. As will be seen, in each case the analytical performance approximation is a consistently good approximation to the simulated error performance.

4.4.1 96-QAM, 80-QAM and 64-QAM Constellations

Figure 4.3 illustrates the multilevel block code performance using three constituent constellations: 96-QAM with a four-dimensional shaping code; 80-QAM with an eight-dimensional shaping code; and 64-QAM with no shaping code. In these examples, we use the Reed-Muller component codes defined as case one in Table 4.1 – $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$ and $C_3 = (32, 26, 4)$.

The overall throughput of each of these multilevel block code constructions is given in Table 4.2. We compare the performance of each case to uncoded 32-QAM, which has a throughput of $\rho = 5$ bits/symbol.

Table 4.2: Throughput, ρ, of the case one component codes.

constellation	ρ (bits/symbol)
96-QAM	5
80-QAM	4.75
64-QAM	4.5

From Figure 4.3, we note that at a BER of 10^{-5} the coding gains are approximately 2.5 dB, 3 dB and 3.25 dB over 32-QAM. The nominal coding gains given in Equation (3.5) [33] are 4.1 dB, 5 dB and 5.8 dB respectively. For each example we substitute the error coefficient for the last non-trivial code, $A_{d_{min},3} = 1240$, and the MSED of the overall code, $D^2_{min}(\mathbf{C}) = 8\delta_0^2$, into Equation (4.5). As shown in this Figure, the approximation of Equation (4.5) provides excellent estimates of performance at moderate to high SNR.

As the number of signal points in a constellation is increased, the MSED between signal points decreases. In order to maintain a given level of performance when expanding a QAM signal set, the average system power must be increased. For example, the addition of one bit per symbol costs 3-dB in power to sustain a given level of performance [13]. Similarly a 1.5-dB performance loss would be expected by increasing the number of bits per symbol by

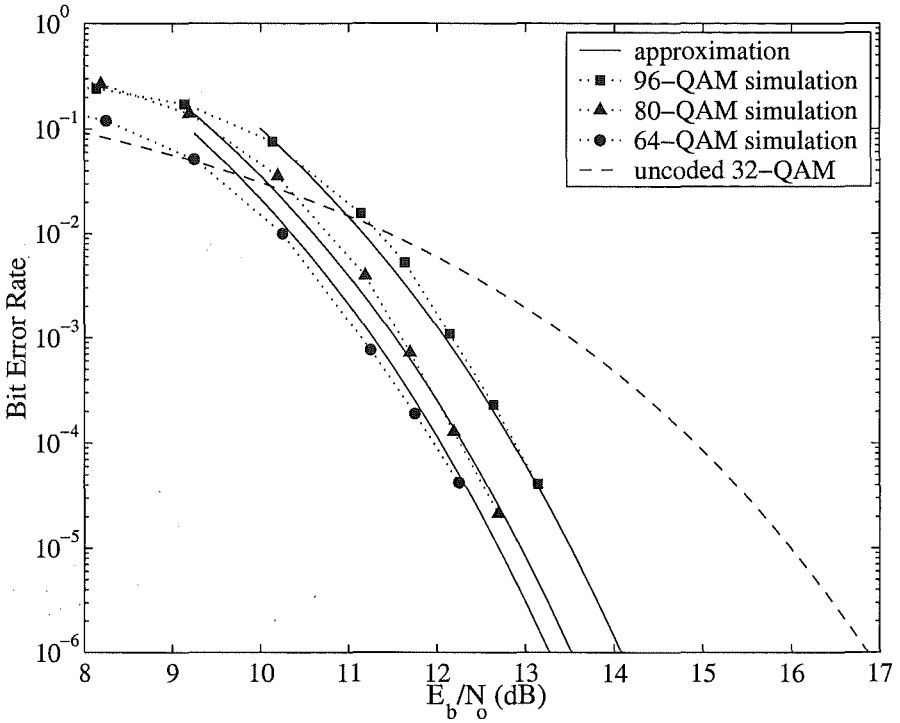


Figure 4.3: Performance of coded 96-QAM, 80-QAM and 64-QAM multilevel block codes utilising Reed-Muller component codes versus uncoded 32-QAM.

half a bit per dimension – that is, increasing the constellation size from 64-QAM to 96-QAM. Figure 4.3 shows that at a BER of 10^{-5} the performance of coded 64-QAM is approximately 0.75 dB better than that of coded 96-QAM. This indicates that approximately half the performance loss due to signal set expansion has been recovered through the shaping code reducing the PAR.

Figure 4.4 illustrates the performance of four-dimensional coded 96-

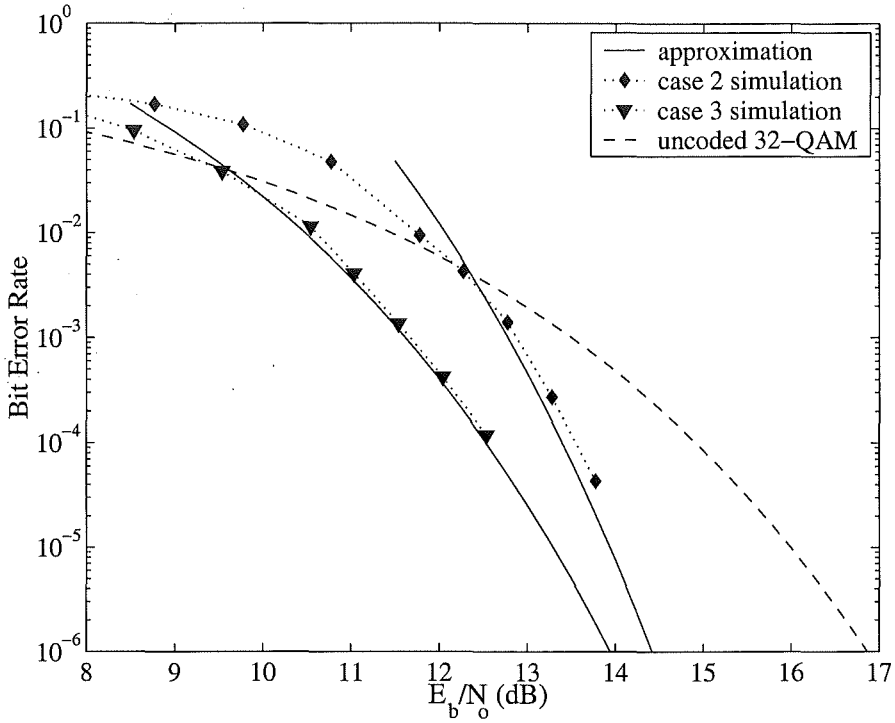


Figure 4.4: Performance of two coded 96-QAM multilevel block code codes utilising different hierarchies of Reed-Muller component codes.

QAM with different sets of Reed-Muller component codes. These examples use the hierarchy of component codes defined as cases two and three in Table 4.1 – the respective error coefficients of C_3 are also shown in this Table. The throughput in each of these cases for 96-QAM is given in Table 4.3. Here,

Table 4.3: Throughput, ρ (bits/symbol), of 96-QAM, 80-QAM and 64-QAM using component codes variations.

constellation	case two	case three
96-QAM	$\rho = 5.4$	$\rho = 4.56$
80-QAM	$\rho = 5.19$	$\rho = 4.3$
64-QAM	$\rho = 4.93$	$\rho = 4.06$

although the MSED of each code is $D^2_{min}(\mathbf{C}) = 8\delta_0^2$, the error performance at moderate SNR varies depending on $A_{d_{min},3}$. For example, at a BER of 10^{-4}, case two has a performance improvement of approximately 1.25 dB over uncoded 32-QAM, whereas case three exhibits approximately 2.5-dB of improvement over uncoded 32-QAM. However, the two curves approach each other as SNR increases. This indicates that at high SNR the size of the error coefficient is not important and at low SNR a very large error coefficient will degrade overall performance. Referring to Equation (3.5), the nominal coding gain for both these examples is 4.1 dB over uncoded 32-QAM. Again, the approximation to the error performance provides a good estimation of the simulated performance.

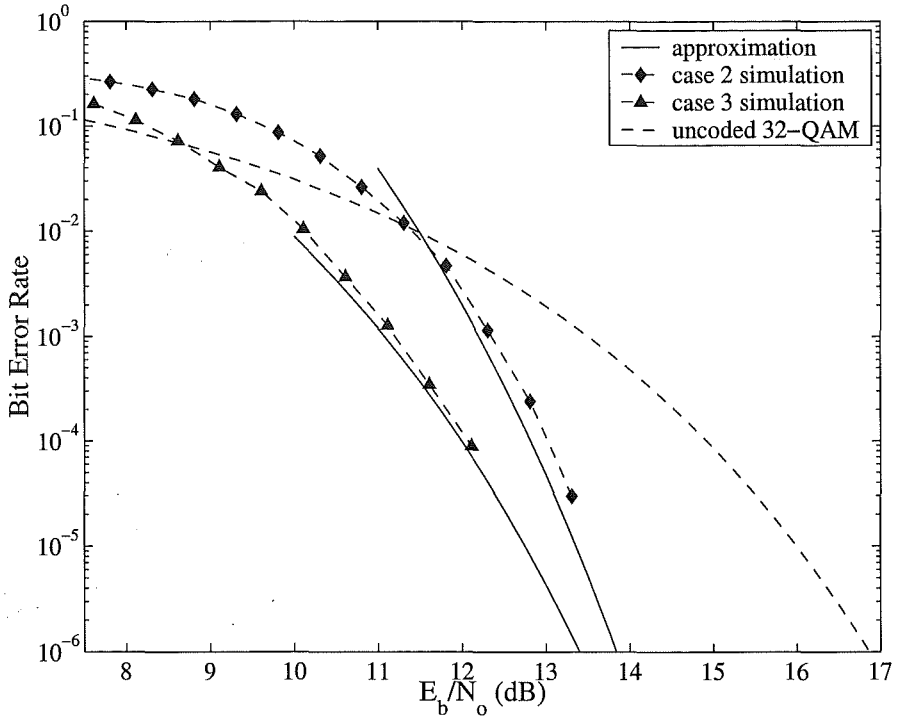


Figure 4.5: Performance of two coded 80-QAM multilevel block code codes utilising different hierarchies of Reed-Muller component codes.

In Figures 4.5 and 4.6, we illustrate the performance of 80-QAM and 64-QAM using the two hierarchies of component codes defined as case two and three in Table 4.1. The respective throughput that these multilevel codes have is given in Table 4.3. In all four examples, the analytical performance is a close approximation to the simulated results. Again, in both 80-QAM and 64-QAM, case three has better performance at low-to-moderate values of SNR.

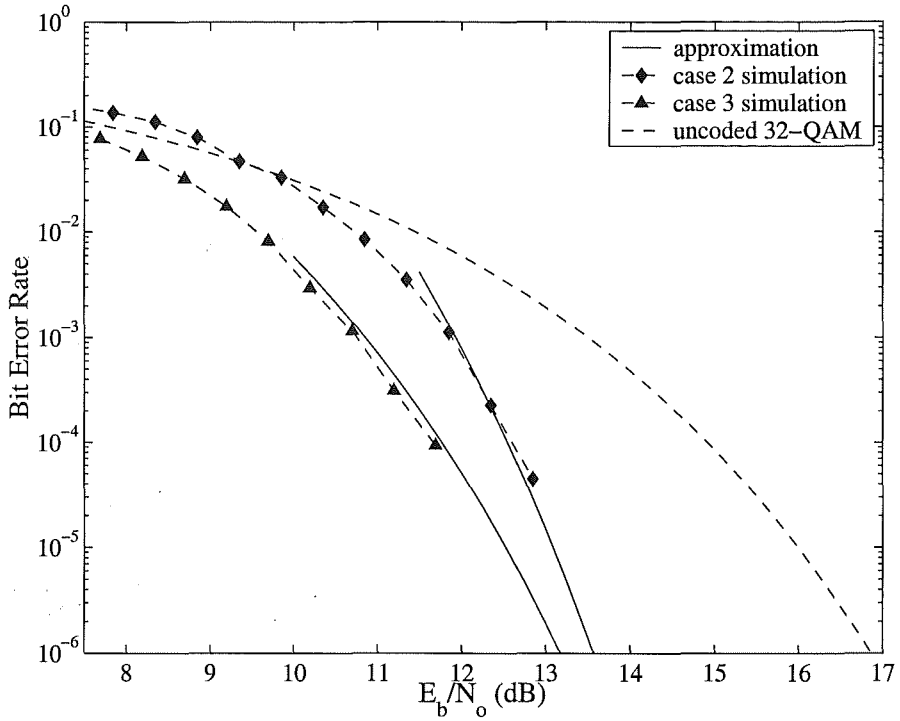


Figure 4.6: Performance of two coded 64-QAM multilevel block code codes utilising different hierarchies of Reed-Muller component codes.

This demonstrates that the error coefficient of the last non-trivial code, $A_{d_{min},3}$, dominates performance, even though the MSED of each code is $D^2_{min}(\mathbf{C}) = 8\delta_0^{22}$; however, the performance in both examples, again, approaches that of case two as SNR increases.

[2]The path structure of the last non-trivial code determines the path structure of and, therefore, the number of nearest neighbours between clusters of parallel transitions.

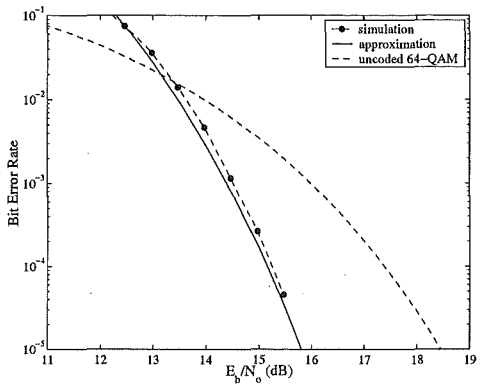
4.4.2 192-QAM and 128-QAM Constellations

One approach to increase throughput is to add another parallel transition to the coded system. As examples, we add a parallel transition to the 96-QAM and 64-QAM constellations described in the previous Section to achieve 192-QAM and 128-QAM respectively. Coded 192-QAM utilises a four-dimensional shaping code. Coded 128-QAM has no shaping code associated with it since it is a two-dimensional constellation. We simulate these constellations using the three cases of Reed-Muller component code hierarchies given in Table 4.1. The respective throughputs for each example of coded 192-QAM and coded 128-QAM are given in Table 4.4. The reference constellation used here is uncoded 64-QAM which has a throughput of $\rho = 6$ bits/symbol.

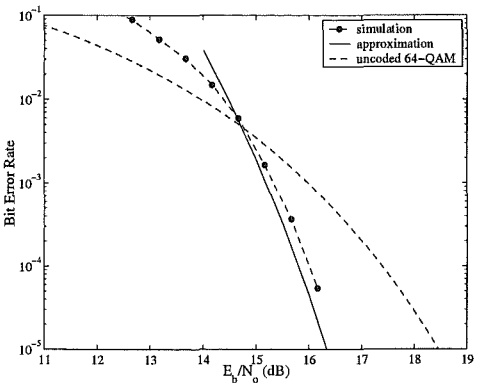
Table 4.4: Throughput, ρ (bits/symbol), of 192-QAM and 128-QAM using component codes variations.

constellation	case one	case two	case three
192-QAM	$\rho = 6$	$\rho = 6.4$	$\rho = 5.56$
128-QAM	$\rho = 5.5$	$\rho = 5.93$	$\rho = 5.06$

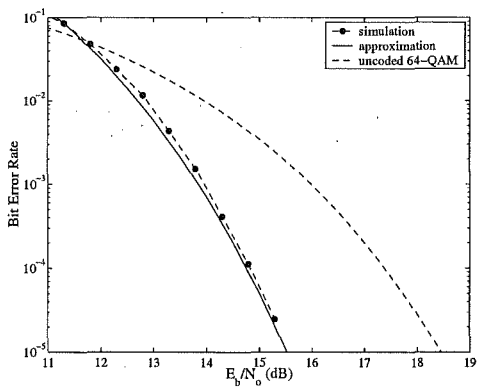
In Figures 4.7(a)-(c) the 192-QAM examples are illustrated. Its simulated coding gain at a BER of 10^{-4} for each case is approximately 2.15 dB, 1.4 dB and 2.65 dB respectively over uncoded 64-QAM. Figures 4.7(d)-(f) display the performance of coded 128-QAM. Here the simulated coding gain in



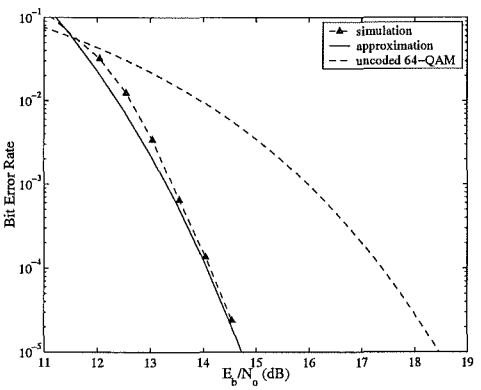
(a) 192-QAM case one



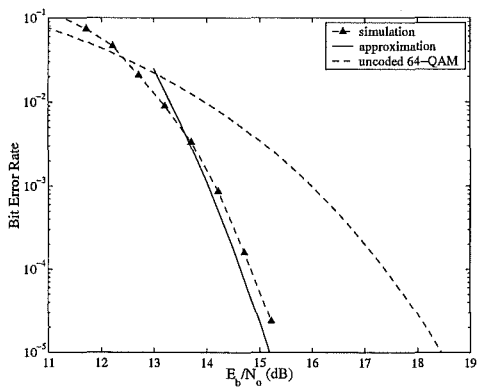
(b) 192-QAM case two



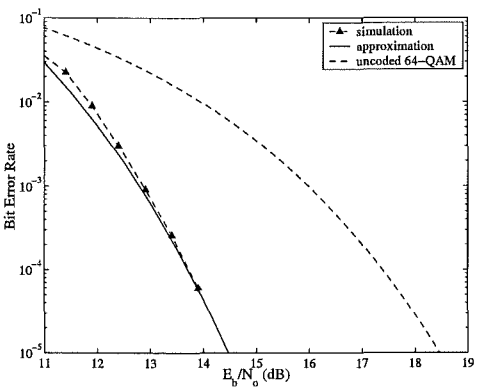
(c) 192-QAM case three



(d) 128-QAM case one



(e) 128-QAM case two



(f) 128-QAM case three

Figure 4.7: Coded 192-QAM and 128-QAM examples using various component codes combinations. Performance is compared to uncoded 64-QAM.

each case at a BER of 10^{-4} is approximately 3.25 dB, 2.5 dB and 3.5 dB respectively over uncoded 64-QAM. These Figures all show Equation (4.5) to be a good analytical approximation to simulated performance for systems using larger QAM constellations.

In terms of the loss in performance when increasing the constellation sizes one bit per symbol (from 64-QAM to 128-QAM, or 96-QAM to 192-QAM), there is an approximate 3 dB loss in asymptotic SNR performance. More specifically, at a BER of 10^{-4} the performance loss when increasing the constellation size by one bit per symbol is approximately 2.5 dB in both cases. However, in the coded cases when increasing the constellation sizes from 64-QAM to 128-QAM, the loss in performance at a BER of 10^{-4} is about 2 dB and from 96-QAM to 192-QAM the loss in performance is approximately 2.25 dB. That is, when adding an extra bit per symbol, the loss in performance is larger in the uncoded case than it is when incorporating a code.

4.4.3 48-QAM and 32-QAM Constellations

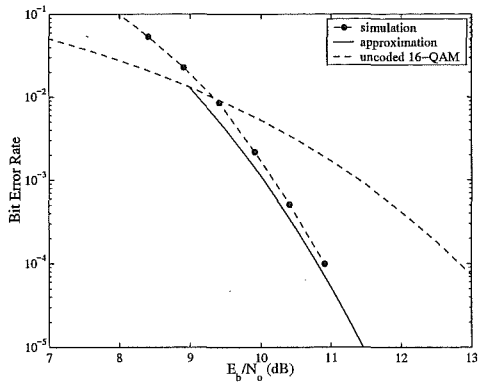
We now illustrate that the same approximations and expansions are true for smaller constellations than those previously considered. As examples, we now remove a parallel transition from the original 96-QAM and 64-QAM examples to give 48-QAM and 32-QAM respectively. We now have a total of two parallel transitions for each constellation. Associated with coded 48-QAM is a four-dimensional shaping code and in these cases, 32-QAM is not shaped. We simulate each constellation using the three alternative cases of Reed-Muller component code combinations given in Table 4.1. The respective throughputs

for the 48-QAM and 32-QAM constellations are given in Table 4.5. The reference constellation used here is uncoded 16-QAM which has a throughput of $\rho = 4$ bits/symbol.

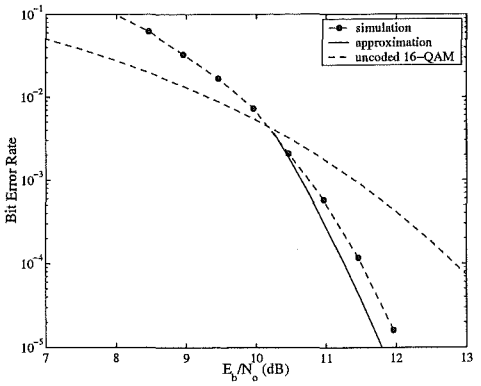
Table 4.5: Throughput, ρ (bits/symbol), of 48-QAM and 32-QAM using component codes variations.

constellation	case one	case two	case three
48-QAM	$\rho = 4$	$\rho = 4.4$	$\rho = 3.56$
32-QAM	$\rho = 3.5$	$\rho = 3.94$	$\rho = 3.06$

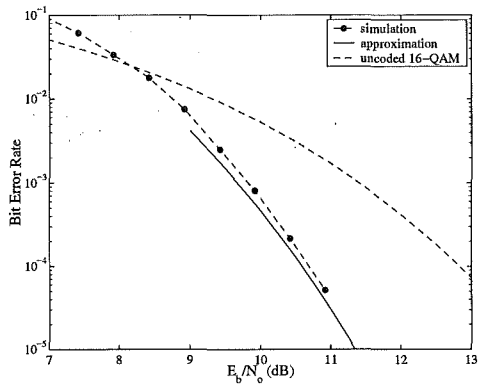
As may be seen in Figures 4.8(a)-(c), the simulated coding gain of each coded 48-QAM case at a BER of 10^{-4} is approximately 2 dB, 1.25 dB and 2.2 dB respectively over 16-QAM. Figures 4.8(d)-(f) show the 32-QAM cases where the coding gains at a BER of 10^{-4} are approximately 2.75 dB, 2.25 dB and 3 dB respectively over 16-QAM. In all these examples the performance curves show that Equation (4.5) remains a good analytical approximation to the simulated code performance when using smaller constellations.



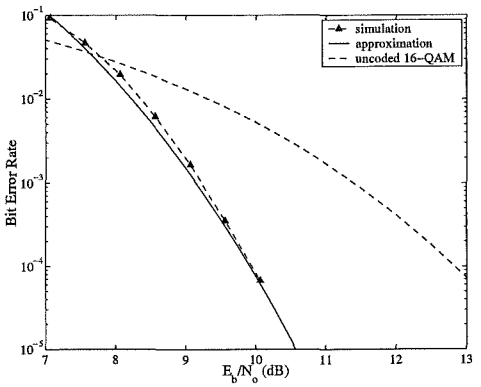
(a) 48-QAM – case one



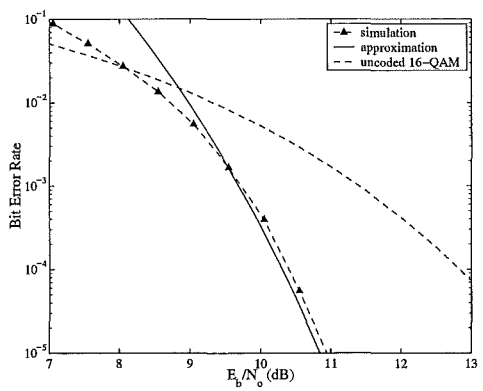
(b) 48-QAM – case two



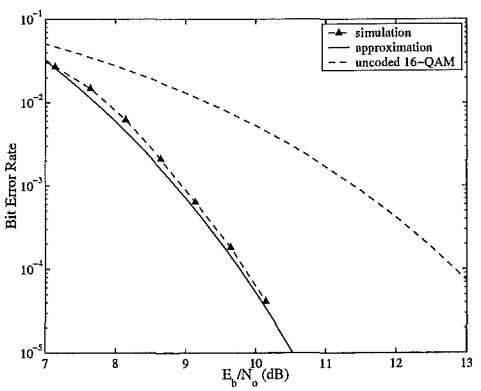
(c) 48-QAM – case three



(d) 32-QAM – case one



(e) 32-QAM – case two



(f) 32-QAM – case three

Figure 4.8: Coded 48-QAM and 32-QAM examples using various component codes combinations. Coded performance is compared to uncoded 16-QAM.

4.4.4 24-QAM and 16-QAM Constellations

Finally, we compare the analytical approximation to the simulated results for three alternative component code cases for coded 24-QAM and coded 16-QAM constellations. In this instance, we are removing two parallel transitions from the original 96-QAM and 64-QAM examples – this is equivalent to removing one parallel transition from the 48-QAM and 32-QAM examples. The 24-QAM and 16-QAM constellations each have one parallel transition – that is, one uncoded bit. Here, coded 24-QAM has a four-dimensional shaping code associated with it and 16-QAM has no shaping code. Each constellation is simulated using the three cases of Reed-Muller component code hierarchies. The respective throughputs for the 24-QAM and 16-QAM constellations are given in Table 4.6. As one may observe, each of these throughputs are simply one bit per symbol less that the 48-QAM and 32-QAM examples as is expected. The reference constellation used here is uncoded 4-QAM which has a throughput of $\rho = 2$ bits/symbol.

Table 4.6: Throughput, ρ (bits/symbol), of 24-QAM and 16-QAM using component codes variations.

constellation	case one	case two	case three
24-QAM	$\rho = 3$	$\rho = 3.4$	$\rho = 2.56$
16-QAM	$\rho = 2.5$	$\rho = 2.94$	$\rho = 2.06$

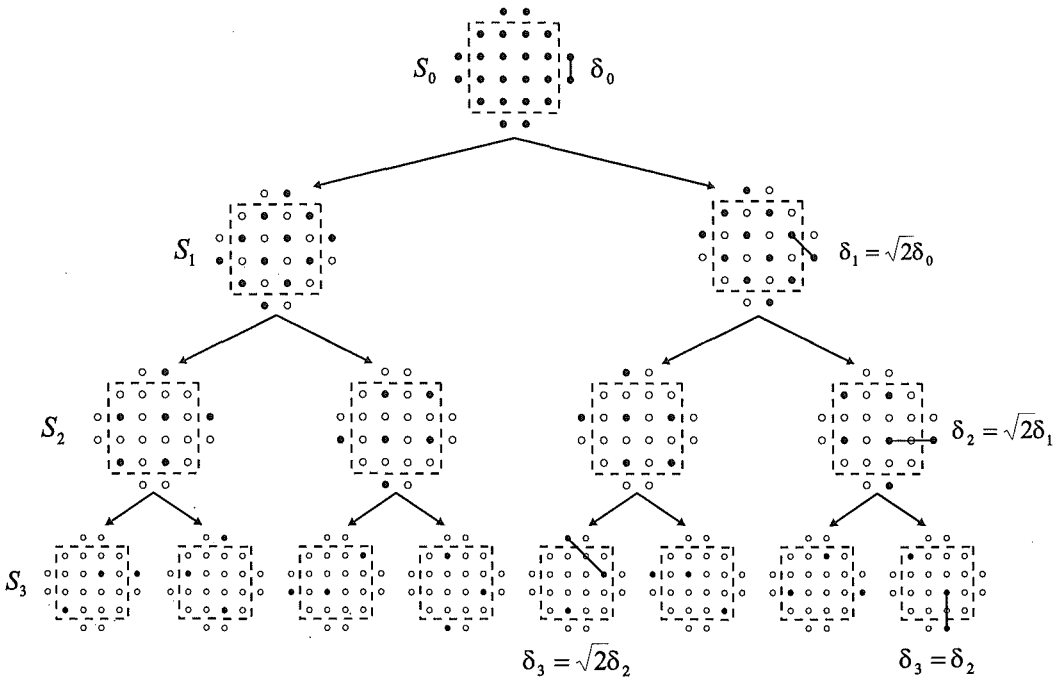


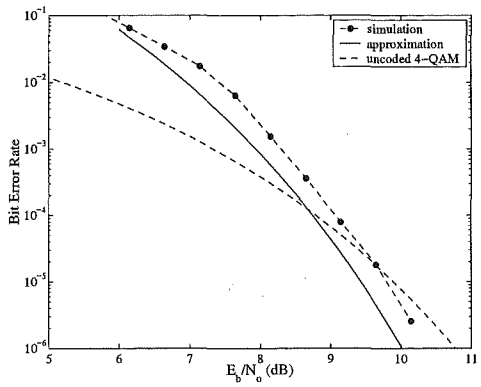
Figure 4.9: Ungerboeck set partitioning of 24-QAM.

We will initially deal with the 24-QAM situation since there is a slight anomaly with the constellation partitioning. This anomaly is reflected in the analytical versus simulated performance curves. Referring to Figure 4.9, one may observe that the constituent constellation may be successfully partitioned twice with increasing MSED between signal points in subsets as specified in Equation (3.2). The first, second and third partitions have inter-subset minimum squared Euclidean distances of δ_0^2, $\delta_1^2 = 2\delta_0^2$ and $\delta_2^2 = 2\delta_1^2$ respectively. In each of these partition levels there are equal numbers of inner points in each

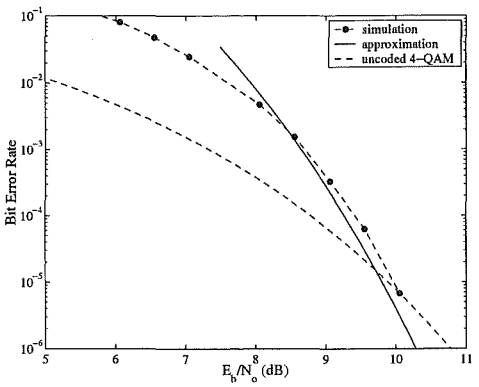
subset and equal numbers of outer points. However, in the third partition the intra-subset MSED is $\delta_3^2 = 2\delta_2^2$ in half the subsets and $\delta_3^2 = \delta_2^2$ in the remaining subsets. Therefore, the requirement for increasing Euclidean distance at each subset level is not met. This is a problem with some smaller constellations. However, in the subsets where the intra-subset MSED is not increasing, the offending signal points relates to the MSED between an inner and an outer point. Otherwise the inner points all have a MSED of $\delta_3^2 = 2\delta_2^2$.

Since we are using a shaping code, the number of times we may use the offending outer point is restricted by the shaping. In addition, the shaping code may offer a small error correcting capability in its decoding algorithm – since if more than one outer point is received, the shaping decoder must correct for that by choosing the most likely inner point in place of the erroneous outer point. In these 24-QAM examples, we continue to use the hierarchy of three component codes with decreasing Hamming distance as in the previous examples. However, the simulated performance is a little worse than if the MSED between signal points in subsets increased uniformly. In the analytical approximation to the error performance, we use the error coefficient of the last non-trivial codes and as the MSED of the code we substitute $D_{min}^2(\mathbf{C}) = 2\delta_2^2$.

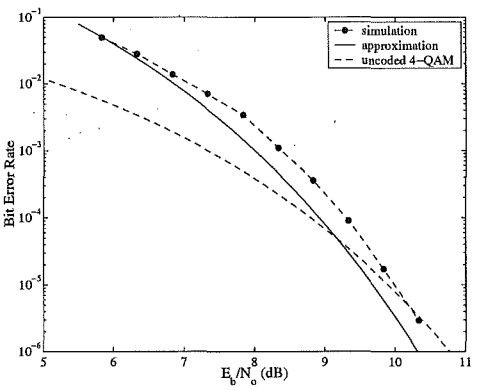
As we may observe in Figures 4.10(a)-(c), the simulated performance is a little worse than the analytical approximation as expected. The simulated performance of each coded 24-QAM constellation is approximately equal to the performance of uncoded 4-QAM at a BER of 10^{-5}. On the other hand, in the coded 16-QAM case, each signal set partition has increasing MSED as is required. The analytical approximation and the simulated results are in good



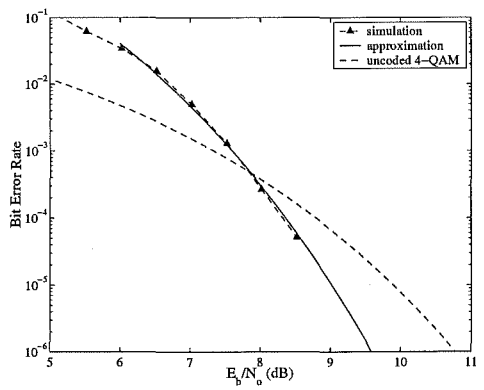
(a) 24-QAM – case one



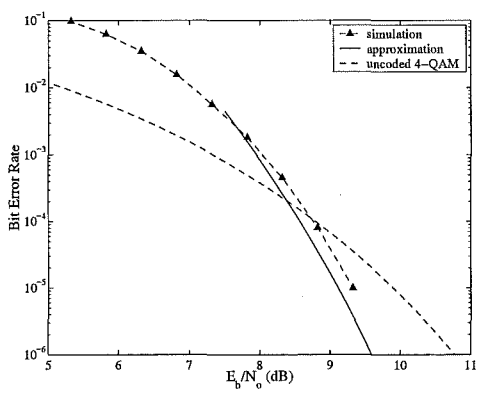
(b) 24-QAM – case two



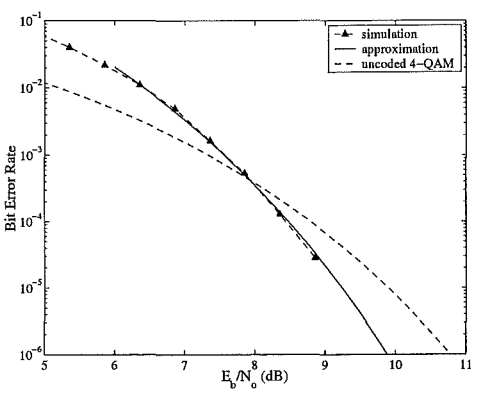
(c) 24-QAM – case three



(d) 16-QAM – case one



(e) 16-QAM – case two



(f) 16-QAM – case three

Figure 4.10: Coded 24-QAM and 16-QAM examples using various component codes combinations. Coded performance is compared to uncoded 4-QAM.

agreement as shown in Figures 4.10(d)-(f). The coding gains in each case for the coded 16-QAM constellation are approximately 0.4 dB, 0.2 dB and 0.3 dB respectively over 4-QAM at a BER of 10^{-4}. Note, because the throughput in the coded 24-QAM and coded 16-QAM examples varies between $\rho = 2$ bits/symbol and $\rho = 3$ bits/symbol, 8AM-PM – which has a throughput of ρ =3 bits/symbol – would also be a good modulation to compare performance. This would make the relative performance of these two coded cases appear better.

4.5 Summary

This Chapter has considered an analytical approximation to the error performance of multilevel block coded modulation schemes on an AWGN channel. The analytical approximation is compared to computer simulated results. As examples, we considered several multilevel block coded schemes based on various QAM constellations and three alternative Reed-Muller component code hierarchies. In each case, as the size of the QAM constellation increases the analytical results are a tighter approximation to the simulated results. On the other hand, when the set partitioning of smaller constellations is restricted, such that increasing MSED is not achievable in all subsets, the analytical and simulated performances are not as close. The analytical results show that the error coefficient influences performance. In the present instance, this has meant allowing the parallel transitions to dominate the exponential behaviour in order to obtain a reduced error coefficient.

Chapter 5

COMBINED EQUALISATION AND DECODING

The road to wisdom? - Well, it's plain
and simple to express:
Err and err and err again
but less and less and less
Piet Hein

5.1 Introduction

This Chapter develops a simple iterative approach to combined equalisation and multilevel block decoding. It follows the Baysian DFE concepts of [39, 40] and the joint equalisation and decoding techniques of [40, 41]. Essentially, we combine decision-feedback equalisation [37] and the decoding of multilevel block codes on large QAM constellations [31, 33, 34] through the use of a simple iterative technique [42]. Following [43], it is clear that the forward and feedback filters of the DFE may be optimised independently. Assum-

ing the optimal forward filter, we may then combine the feedback filter and the decoder in an iterative structure. The decoder output for each received block of symbols is iteratively re-encoded to the original symbol structure and processed through the combination of the DFE feedback filter and the block decoding algorithm. The operation of a system using an iterative process may be thought of in terms of the improvement of information that each stage (or iteration) adds to its overall performance [98].

This Chapter initially reviews conventional, concatenated equalisation and decoding. These are the most commonly implemented structures when a system requires both equalisation and decoding [41, 98]. We point out the draw-backs of these structures which lead us to the development and discussion of the iterative approach to combined equalisation and decoding. We analytically compare and contrast performance expectations of the conventional and combined structures. The decoder used in these schemes is the staged decoder of [31, 33, 34] as discussed in Chapter 3. Here we assume a common block length, n_c, for each component code.

5.2 Conventional Equalisation and Decoding

Conventional equalisation and decoding is commonly implemented by concatenating an equaliser, in this case a DFE, with an error correcting decoder. This is most easily achieved by using the DFE quantiser output, $\hat{x}_i$, as the decoder input [41] (cf. Figure 5.1). The inputs to the decoder are then hard decisions, and therefore, the error correcting code is hard-decision decoded. This method of concatenating equalisation and decoding is a stan-

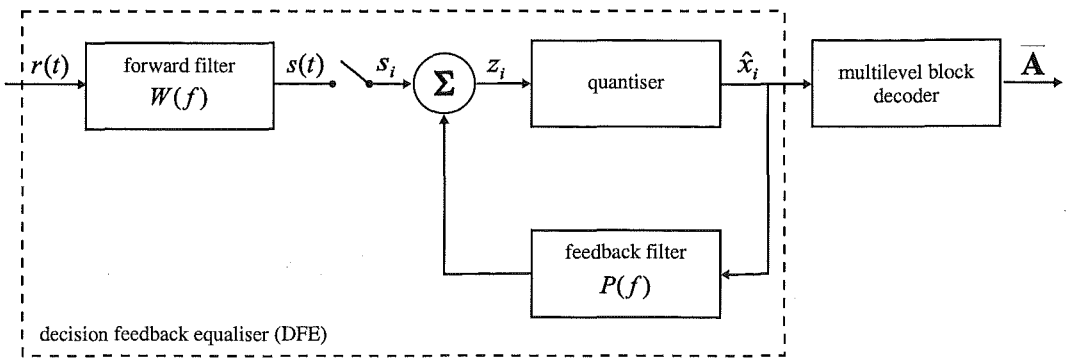


Figure 5.1: Block diagram of conventional, concatenated equalisation and decoding implementing hard-decision decoding.

dard structure used in many systems – it will be referred to as conventional equalisation and hard-decision decoding.

Intuitively, we would expect performance of the conventional equalisation and hard-decision decoding system to be improved if the decoder performs soft-decision decoding[1]. By allowing the decoder access to the estimated degree of confidence associated with the incoming decision, it may perform soft-decision decoding [98]. This is achieved by using the unquantised symbol estimates, z_i, as the decoder input (cf. Figure 5.2). We will refer to this structure as the conventional equalisation and decoding system and will use its performance as a standard of comparison. It is a standard equalisation and decoding structure when soft-decision decoding is desired. It is this structure

[1]Here, soft-decision decoding refers to soft-input decoding.

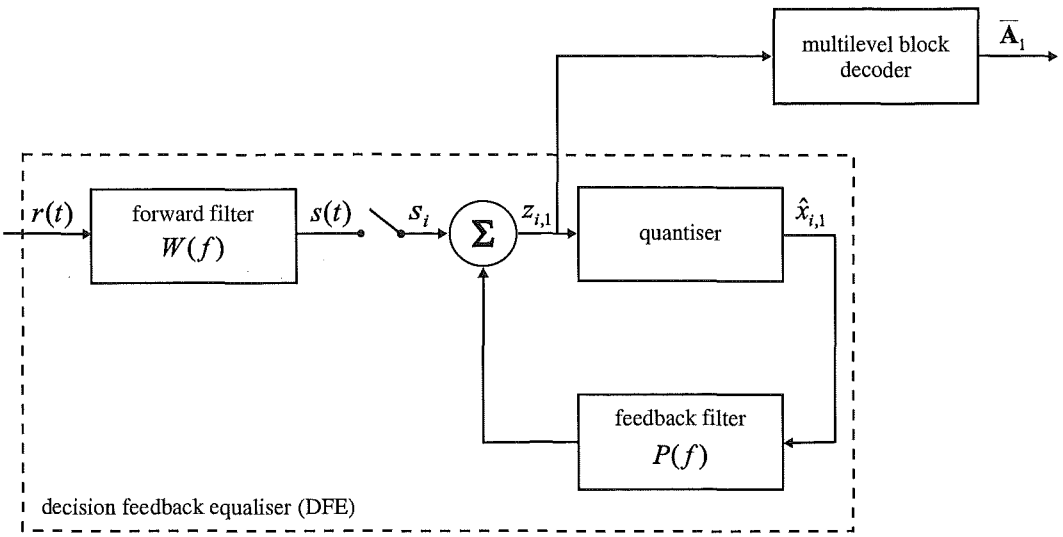


Figure 5.2: Block diagram of conventional, concatenated equalisation and decoding implementing soft-decision decoding.

we use as a basis for combined equalisation and decoding.

When using the conventional equalisation and decoding system, at time $t = iT$ where i is the discrete time index with $i = (0, 1, 2, \cdots)$, the DFE makes an unquantised symbol estimate as in Equation (2.27). It is reproduced here incorporating the notation for the iterative structure

$$z_{i,1} = s_i + \hat{\mathbf{X}}_{i,1}\mathbf{P}_{i,1}^H \tag{5.1}$$

In this case, the subscript "1" is used to indicate the first iteration (or pass) through the system. This allows for the later extension of Equation (5.1) to iterative combined equalisation and decoding. In the i^{th} sampling interval, the

symbol estimate, $z_{i,1}$, is based on the output of the (optimal) forward filter, s_i, such that

$$s_i = \mathbf{R}_i \mathbf{W}_i^H \tag{5.2}$$

At each symbol time the quantiser makes a hard decision, $\hat{x}_{i,1}$, on the estimated symbol, $z_{i,1}$. The symbol values $\hat{x}_{i,1}$ are assumed to be correct and are shifted into the feedback filter. The residual error, $e_{i,1}$, is calculated as in Equation (2.28) such that

$$e_{i,1} = \hat{x}_{i,1} - z_{i,1} \tag{5.3}$$

The forward and feedback filter tap weights are updated according to

$$\mathbf{W}_{i+1} = \mathbf{W}_i + \mu e_{i,1}^* \mathbf{R}_i \tag{5.4}$$

$$\mathbf{P}_{i+1,1} = \mathbf{P}_{i,1} + \mu e_{i,1}^* \hat{\mathbf{X}}_{i,1} \tag{5.5}$$

As will be seen, the forward filter does not enter the iterative process. Following the initial training it is thus adaptively updated only during the first iteration on each block of n_c received symbols. It is otherwise assumed to be essentially at its optimum operating point. The feedback filter, as part of the iterative equalisation/decoding algorithm, is adaptively updated during each iteration.

Each "soft" or unquantised symbol estimate, $z_{i,1}$, is passed to the multilevel block decoder. These values are stored by the decoder until a block of size n_c, corresponding to a complete code word, has been received. The block is then decoded to produce the binary data block $\bar{\mathbf{A}}_1 = [\bar{a}_{1,1}, \bar{a}_{2,1}, \ldots, \bar{a}_{n_d,1}]$,

where n_d is the size of the transmitted binary data block such that $n_d = \sum_{\nu=1}^{m} k_\nu$. In a non-iterated system, $\bar{\mathbf{A}}_1$ is the final output.

Although the error correcting code may be soft-decision decoded in the conventional equalisation and decoding system, the accuracy of the data being fed back in the DFE relies on the quantiser outputs, $\hat{x}_{i,1}$, being correct. Errored symbols in the feedback filter will affect the estimated symbol, $z_{i,1}$; however, provided they are relatively few in number, many of these errors will be corrected by the decoding algorithm.

In a decision-feedback structure, the quantiser outputs, $\hat{x}_{i,1}$, may be regarded [39, 103] as a high-SNR approximation to the optimum or Bayesian minimum mean squared error (MMSE) estimate of the transmitted symbols. In practice, the DFE seeks to minimise[2]

$$E\left[\left|e_{i,1}\right|^2\right] = E\left[\left|\hat{x}_{i,1} - z_{i,1}\right|^2\right] \rightarrow min \tag{5.6}$$

where $E[\bullet]$ denotes expected value; and $e_{i,1}$ is the residual error from Equation (5.3). Thus we may, conceptually at least, replace the quantiser with the MMSE estimator [40, 103]

$$\tilde{x}_{i,1} = E\left[x_i \middle| z_{i,1}\right] = \sum_{\forall x_i} x_i P\left(x_i \middle| z_{i,1}\right) \tag{5.7}$$

where x_i is the actual symbol at time i; and $P(x_i|z_{i,1})$ is the *a posteriori* probability of x_i given $z_{i,1}$ and may be expressed as

$$P\left(x_i \middle| z_{i,1}\right) = \frac{P\left(x_i, z_{i,1}\right)}{P\left(z_{i,1}\right)} = \frac{P\left(z_{i,1} \middle| x_i\right) P\left(x_i\right)}{\sum_{\forall x_i} P\left(z_{i,1} \middle| x_i\right) P\left(x_i\right)} \tag{5.8}$$

[2]In most analyses of DFE behaviour, the quantiser outputs, $\hat{x}_{i,1}$, are assumed to be correct.

It is easily shown [40, 103, 104] that Equation (5.7) is a soft quantiser characteristic which approaches hard quantisation at high SNR. In fact, for the binary case it results in the estimator

$$\tilde{x}_{i,1} = \tanh\left(\frac{z_{i,1}}{\sigma_n^2}\right) \tag{5.9}$$

At high SNR ($\sigma_n^2 \rightarrow 0$), this clearly becomes a hard binary quantiser. A similar conclusion may easily be reached in the case of M-ary symbols x_i, so that at high SNR the estimates $\tilde{x}_{i,1}$ approach the hard quantiser outputs, $\hat{x}_{i,1}$. It is the *a posteriori* probability based estimator of Equation (5.7) that motivates the simple iterative processor of this Thesis, in that a MMSE symbol will on average be more reliable than a hard quantiser decision. In order to keep the traditional DFE structure we feed back decisions from the decoder instead of a MMSE estimator.

5.3 Combined Equalisation and Decoding

A drawback of conventional equalisation is that the DFE suffers from error propagation in the feedback filter. This error propagation is not catastrophic – it is self-correcting after a few symbols – however, the errors that do occur affect future decisions [13, 40, 41, 65]. This limits the achievable coding gain in the conventional equalisation and decoding systems of Figures 5.1 and 5.2. The limitation is largely due to the fact that errors occurring as a result of error propagation in the DFE are bursty in nature [40], whereas error correcting codes are designed to correct more randomly occurring errors.

In general, many of the errors in the estimated symbols, $z_{i,1}$, will be

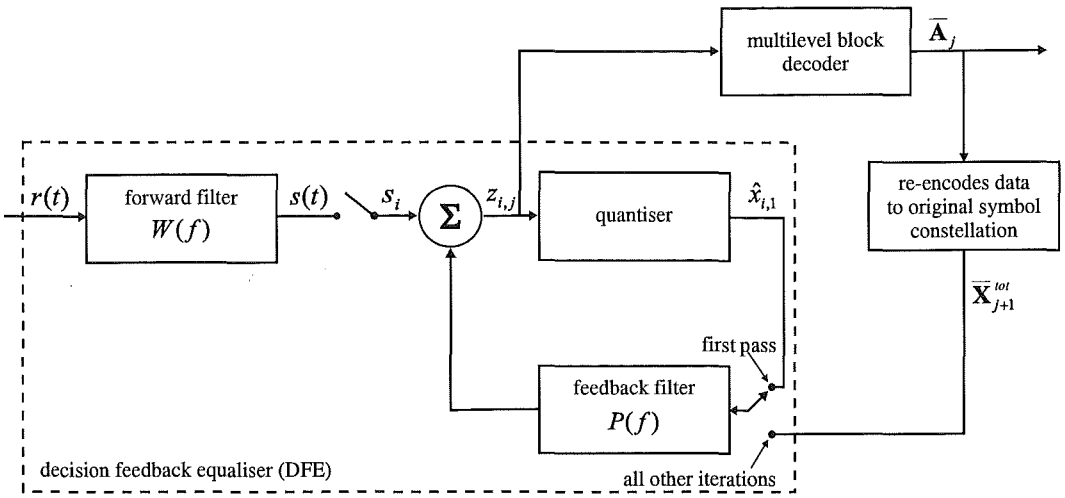


Figure 5.3: Block diagram of an iterative combined DFE and multilevel block decoding structure.

corrected by the decoder. To further improve system performance, we may use the decoder output to construct the input to the feedback filter instead of using the hard decisions from the quantiser. To do this, each decoded data block is re-encoded and regenerated into a sequence of symbol levels. These are then used in the DFE feedback filter in place of the quantiser outputs to re-process the received signal and to re-decode each code block. This process may be repeated several times and results in a simple iterative equalisation and decoding system as shown in Figure 5.3 (switch in lower position). The process has been found to improve system performance, which is then determined in part by the number of equalisation and decoding iterations on each block of

received symbols. We may view the feedback filter in the DFE as a one path approximation to the Viterbi Algorithm[3]. By making these extra iterations, we are essentially making this one path approximation more accurate.

During the initial iteration on each block of n_c symbols, the combined equalisation/decoding system operates as a conventional equalisation/decoding system (cf. Figure 5.3 - switch as shown), and the quantiser outputs, $\hat{x}_{i,1}$, are used in the feedback filter. In the second iteration, each block of decoded data, $\bar{\mathbf{A}}_1 = [\bar{a}_{1,1}, \bar{a}_{2,1}, \ldots, \bar{a}_{n_d,1}]$, is re-encoded and regenerated to symbols using the same encoder and constellation mapping as in the transmitter. The re-encoded symbol block, $\bar{\mathbf{X}}_2^{tot} = [\bar{x}_{1,2}, \bar{x}_{2,2}, \ldots, \bar{x}_{n_c,2}]$, forms a more reliable estimate of the block of transmitted symbol levels than the original quantiser output. Equalisation of the block is then repeated using the original received signal samples and the final estimates of the feedback filter contents from the start of the last block, $\hat{\mathbf{X}}_{i,1}$, which have been stored. The contents of $\hat{\mathbf{X}}_{i,1}$ are then replaced in turn by the decoded symbols in $\bar{\mathbf{X}}_2^{tot}$ giving $\bar{\mathbf{X}}_{i,2}$.

The unquantised or soft symbol estimates during the second iteration, $z_{i,2}$, are calculated as

$$z_{i,2} = s_i + \bar{\mathbf{X}}_{i,2}\mathbf{P}_{i,2}^H \tag{5.10}$$

where s_i is the original output of the (optimal) forward filter at time $t = iT$ as in Equation (5.2); and $\mathbf{P}_{i,2}$ is the row vector of the feedback tap weights during the second iteration. The feedback tap weights are adaptively updated such that at the start of the second iteration $\mathbf{P}_{i,2} = \mathbf{P}_{i+n_c-1,1}$, $(i = 0)$. The

[3]Optimal equaliser structures can be shown to implement maximum likelihood sequence estimation (MLSE) via the Viterbi Algorithm [75]

sequence of estimates, $z_{i,2}$, of Equation (5.10) is passed to the decoder and re-decoded to produce the new data block $\bar{\mathbf{A}}_2 = [\bar{a}_{1,2}, \bar{a}_{2,2}, \ldots, \bar{a}_{n_d,2}]$.

The residual error corresponding to each symbol may now be calculated by replacing the quantiser outputs with the re-encoded symbol values, $\bar{x}_{i,2}$, to obtain

$$e_{i,2} = \bar{x}_{i,2} - z_{i,2} \tag{5.11}$$

which in turn is used in Equation (5.5) to update the feedback tap weights $\mathbf{P}_{i+1,2}$ such that

$$\mathbf{P}_{i+1,2} = \mathbf{P}_{i,2} + \mu e^*_{i,2} \bar{\mathbf{X}}_{i,2} \tag{5.12}$$

The second iteration is complete when all data from $\bar{\mathbf{X}}_2^{tot}$ has been processed. The third iteration re-encodes the data block $\bar{\mathbf{A}}_2$ into $\bar{\mathbf{X}}_3^{tot}$ and repeats the process, which may continue for any number of iterations.

Given the decoded symbols from the first iteration and assuming them correct, the equaliser now seeks to minimise $E[|e_{i,2}|^2]$ [40] such that

$$E\left[|\bar{x}_{i,2} - z_{i,2}|^2 \middle| \bar{\mathbf{X}}_{i,2}\right] \rightarrow min \tag{5.13}$$

where $\bar{x}_{i,2}$ is a function of $\bar{\mathbf{X}}_{i,2}$; and $\bar{\mathbf{X}}_{i,2}$ is the sequence of symbols reconstructed from the decoded data block during the initial pass. The solution to Equation (5.13) is well known and is given as [40]

$$\tilde{x}_{i,2} = E\left[x_i \middle| z_{i,2}, \bar{\mathbf{X}}_{i,2}\right] = \sum_{\forall x_i} x_i P\left(x_i \middle| z_{i,2}, \bar{\mathbf{X}}_{i,2}\right) \tag{5.14}$$

where $\tilde{x}_{i,2}$ is the MMSE estimate of the quantised symbol in the second iteration; and $P(x_i|z_{i,2}, \bar{\mathbf{X}}_{i,2})$ is the *a posteriori* probability of x_i given the remodulated symbol block, $\bar{\mathbf{X}}_{i,2}$, and the unquantised estimates, $z_{i,2}$, during the second

iteration. Equation (5.14) may be shown to be a more reliable estimate of the transmitted symbols than Equation (5.7), and may be shown [40,103,104] to be a soft quantiser characteristic that becomes increasingly hard at high SNR. The estimates $\tilde{x}_{i,2}$ may thus be considered to contain reliability information which is incorporated into the decoder inputs $z_{i,2}$. In fact, since the decoding algorithm is near maximum likelihood, the remodulated symbol block $\bar{\mathbf{X}}_2^{tot}$ provides a sequence of near maximum likelihood estimates. These provide a good approximation to the MMSE estimates of Equation (5.14).

In the second iteration, the unquantised symbol estimates, $z_{i,2}$, of Equation (5.10) are more accurate estimates than the estimates, $z_{i,1}$, of Equation (5.1). Thus, decoding of $z_{i,2}$ yields the decoded data block $\bar{\mathbf{A}}_2$ which on average will contain fewer bit errors than $\bar{\mathbf{A}}_1$. This iterative process may be extended to $I \geq 3$ iterations to improve performance further. Each iteration further refines the data from the previous iteration and we achieve a performance gain each time.

Assuming the filters in the DFE have optimum tap weights at the conclusion of the training sequence, the work of [43] has shown that we may treat the forward filter separately from the feedback filter. The most practicable iterative scheme updates the forward tap weights only on the first pass through each received block of symbols such that the forward filter is not involved in the iterative process. Nominally, we then calculate the symbol estimates in all subsequent iterations as

$$z_{i,j} = s_i + \bar{\mathbf{X}}_{i,j}\mathbf{P}_{i,j}^H \tag{5.15}$$

where $j = (2, \ldots, I)$ and I is the total number of iterations. In practice, we

have found that marginally better performance is obtained by updating the forward tap weights on the first two iterations of each block of n_c symbols to obtain the outputs $s_{i,2}$. Subsequent iterations then use these to obtain the symbol estimates,

$$z_{i,j} = s_{i,2} + \bar{\mathbf{X}}_{i,j}\mathbf{P}_{i,j}^{H} \tag{5.16}$$

where $j = (3, \ldots, I)$. The feedback filter is, of course, adaptively updated during each iteration.

Thus far, the combined equalisation/decoding has been a direct extension from the conventional equalisation/decoding of Figure 5.2. We may apply these methodologies to the conventional equalisation and hard-decision decoding of Figure 5.1 – this is only mentioned in passing as it is a straight-forward modification of the soft-decision decoding case discussed. Its performance, whereby each iteration gives improved performance, is similar in structure to the combined equalisation/decoding which is discussed in Chapter 6. As expected, all the results have a degraded overall performance since no soft-decision decoding is employed.

5.4 Summary

This Chapter has summarised the operation of conventional equalisation and decoding. An approach to combining a DFE and multilevel block decoding based on simple iterative decoding principles is developed and discussed. This approach remodulates symbols from the output of the decoder. Employing these symbols, the received symbol is then re-processed. This operation may

be iterated as many times as desired. We show the proposed structure may be simplified for practical implementation by separating the forward filter from the combined feedback and equalisation process.

Chapter 6

PERFORMANCE EVALUATION: COMBINED EQUALISATION AND DECODING

6.1 Introduction

This Chapter evaluates system performance via simulation of the combined equalisation and decoding algorithm discussed in Chapter 5. We demonstrate here that this combined approach gives a performance gain of up to 1 dB at a BER of 10^{-4} over conventional, concatenated equalisation and soft-decision decoding.

Extensive simulations of the iterative combined equalisation and decoding scheme have been performed. The multilevel block decoding structure used is that of [31, 33, 34] and described in Chapter 3. In all simulations we have used a hierarchy of Reed-Muller component codes, but there is no restriction to their use – other code formats may also be used. Reed-Muller codes provide

a large family of simple codes, are easy to encode and decode and are relatively flexible in correcting a varying number of errors per code word [23]. In addition, they may be easily soft-decision decoded since they have a simple trellis structure [74]. These properties are advantageous for the iterative structure under consideration. In practice, we soft-decision decode only the low level codes C_1 and/or C_2 [34].

To demonstrate the performance of the combined equalisation and decoding structure we have simulated the system using three large constituent QAM constellations – these multilevel block encoded structures are: 64-QAM with no shaping code; 96-QAM with a four-dimensional shaping code; and 80-QAM with an eight-dimensional shaping code [34]. In each example we use the three hierarchies of Reed-Muller component codes defined again in Table 6.1 for convenience.

Table 6.1: Three component code hierarchies, also shown is the error coefficient for C_3.

code hierarchy	C_1	C_2	C_3	$A_{d_{min},3}$
case one	(32,6,16)	(32,16,8)	(32,26,4)	1240
case two	(16,5,8)	(16,11,4)	(16,15,2)	2^{15}
case three	(16,1,16)	(16,5,8)	(16,11,4)	140

The shaping codes are effectively the mapping of parallel transitions to a sequence of multidimensional signal points so as to reduce the trans-

mitted peak-to-average power ratio (PAR) [34]. Unless otherwise stated, in each example C_1 is soft-decision trellis decoded [66] and the remaining component codes are hard-decision decoded using the Reed-Muller decoding algorithm [23]. We assume a two path multipath fading channel. From Equation (2.14), the specific multipath channel chosen for demonstrating system performance has path amplitudes $c_1 = \sqrt{0.54}$, $c_2 = \sqrt{0.46}e^{(-j\pi/1.5)}$ and the relative delay of the second path is set to $\tau_2 = \frac{1}{3}T$. These channel coefficients equate to a 25-dB notch near band edge. Most of the work shown is in this channel; other examples are included later in the Chapter.

The DFE used in this analysis has 16 forward and 16 feedback tap weights. Moreover, the forward filter in all simulations has symbol-spaced taps. By using symbol-spaced taps we make the assumption that the output of the receiver matched filter has perfect sample timing thereby obtaining optimal operation [13, 53]. With a symbol-spaced forward filter, a poor choice of matched filter timing or severe channel characteristics leads to performance degradation due to overwhelming noise enhancement [53]. Perfect sample timing is easily achieved in a simulated situation but is not as straight forward in practice. It is often difficult to design an ideal matched filter with perfect timing since the channel response is not frequently known *a priori.* In practice, a fractionally-spaced forward filter is normally used [13, 53] [1]. It has the useful property that it is able to compensate for filter timing delay without significantly enhancing the additive noise [53].

[1] In both the symbol-spaced and fractionally-spaced decision-feedback equalisers, the feedback filter is symbol-spaced.

6.2 Performance of Conventional Equalisation and Decoding

In Figure 6.1 we illustrate the independent effects of the equaliser and multilevel block code on overall system performance by simulating them in isolation. When using multilevel block coded 64-QAM without equalisation, data is not recovered reliably. Conversely, when using uncoded 64-QAM and equalising the channel, the transmitted data may be reliably recovered at sufficiently high SNR. However, using multilevel block coded 64-QAM with the conventional, concatenated equalisation and decoding system gives a performance gain of more than 3.5 dB at a BER of 10^{-4} over the uncoded 64-QAM example. Furthermore, this gain appears to be increasing as SNR increases. This conventional system represents the system in Figure 5.2 or the first iteration using the structure of Figure 5.3 (switch as shown).

6.3 Performance of 64-QAM, 80-QAM and 96-QAM Constellations with Proposed Combined Equalisation and Decoding

Simulation of the conventional soft-decision decoding system is now extended to the combined iterative equalisation and decoding structure (cf. Figure 5.3 – switch in lower position). As seen in Figure 6.2, system performance improves on each iteration of the combined structure. In this example, we adapt both the forward and feedback filters in the DFE on each itera-

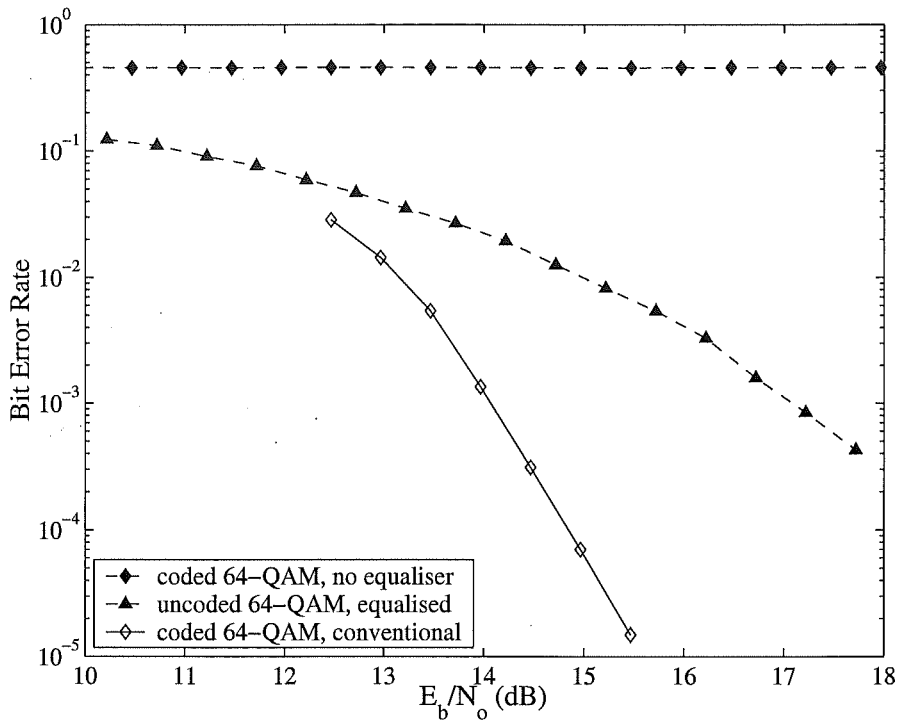


Figure 6.1: Performance of coded 64-QAM with no equalisation; equalised, uncoded 64-QAM; and conventional equalisation and soft-decision decoding of coded 64-QAM. Coded 64-QAM component codes are: $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$, $C_3 = (32, 26, 4)$.

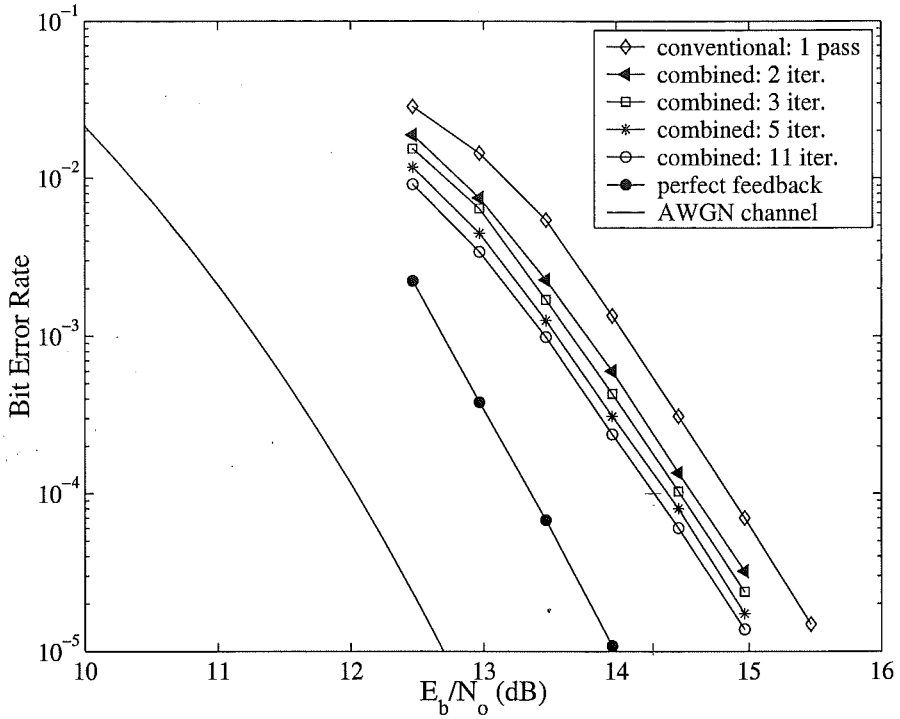


Figure 6.2: Performance of coded 64-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding; performance using perfect DFE feedback; and AWGN channel performance approximation [34]. Component codes: $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$, $C_3 = (32, 26, 4)$.

tion. The conventional system in this Figure refers to the system shown in Figure 5.2. This is also the same performance curve as that in Figure 6.1. As may be observed, when implementing a second iteration, the bit error-rate improves compared to the initial pass by approximately 0.25 dB at a BER of 10^{-4}. Similarly with eleven iterations performance improves compared to one iteration by almost 0.6 dB at a BER of 10^{-4}. From our simulation trials, the performance when using eleven iterations is approaching the maximum achievable gain. The largest performance improvement with the least decoding delay and complexity is achieved by the second iteration. On succeeding iterations, further performance improvement is obtained but with a substantial increase in overall decoder delay. After eleven iterations, adding extra iterations makes a further but fractionally small improvement.

Also illustrated in Figure 6.2 is the simulated performance curve of the conventional equalisation and decoding algorithm when assuming that the symbols fed back into the DFE are exactly the same as those transmitted (perfect feedback). At a BER of 10^{-4}, the performance of perfect feedback is approximately 0.9-dB better than that of the combined equalisation and decoding algorithm using eleven iterations. This shows that in this particular channel, the overall loss in performance when using the conventional system and imperfect decision feedback is approximately 1.5 dB. By utilising the combined system we can recover a little over one third of this performance loss.

Figures 6.3 and 6.4 illustrate the performance of coded 80-QAM and 96-QAM respectively using the same component codes and channel coefficients as in the 64-QAM example. Again, as the number of iterations increases, the

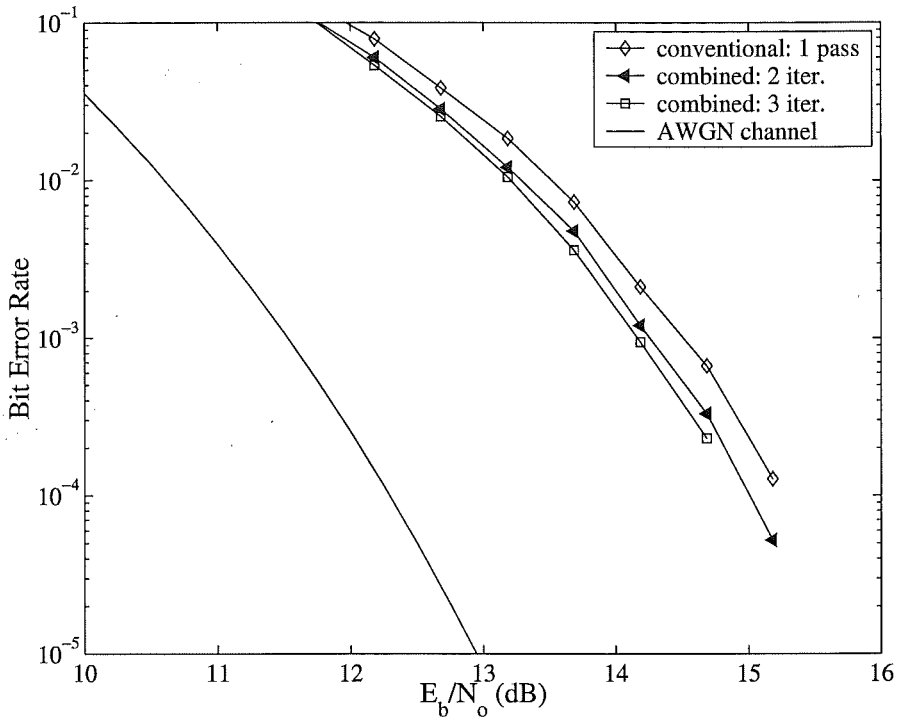


Figure 6.3: Performance of coded 80-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding; and AWGN channel performance approximation [34]. Component codes: $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$, $C_3 = (32, 26, 4)$.

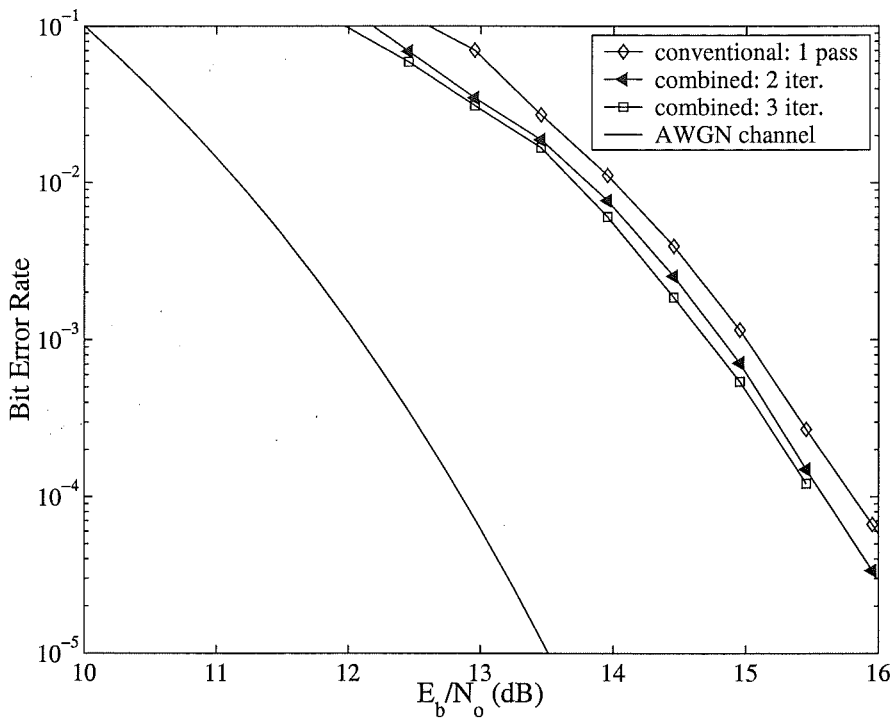


Figure 6.4: Performance of coded 96-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding; and AWGN channel performance approximation [34]. Component codes: $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$, $C_3 = (32, 26, 4)$.

performance improves as indicated by Equation (5.14). Performance of both these systems is slightly worse than the coded 64-QAM case since the MSED between adjacent signal points in these constellations has decreased. This was previously seen through analysis and simulation on the AWGN channel [34].

6.4 Alternative 64-QAM Scenarios using Combined Equalisation and Decoding

We now consider system performance when the multilevel code employs the same number of symbols per block as the number of taps in the feedback filter. As examples, we multilevel encode 64-QAM using two hierarchies of Reed-Muller component codes – namely cases two and three given in Table 6.1. The equaliser used in both cases is the same as in previous examples.

Figures 6.5 and 6.6 show the resulting performance of cases two and three respectively. As in previous simulations, system performance improves as the number of iterations increase. In case two, the performance gain between the first and second iteration is approximately 0.25 dB at a BER of 10^{-4}; and between the second and third iterations there is another small increase. In the hierarchy of codes defined as case three, the performance gain between the first and second iteration is almost 0.5 dB at a BER of 10^{-4}; and between the second and third iterations is approximately 0.15 dB; and between the third and eleventh iteration is a further 0.15 dB.

These simulations suggest that more iterations could tend towards a 1-dB gain over conventional equalisation and decoding. This improved perfor-

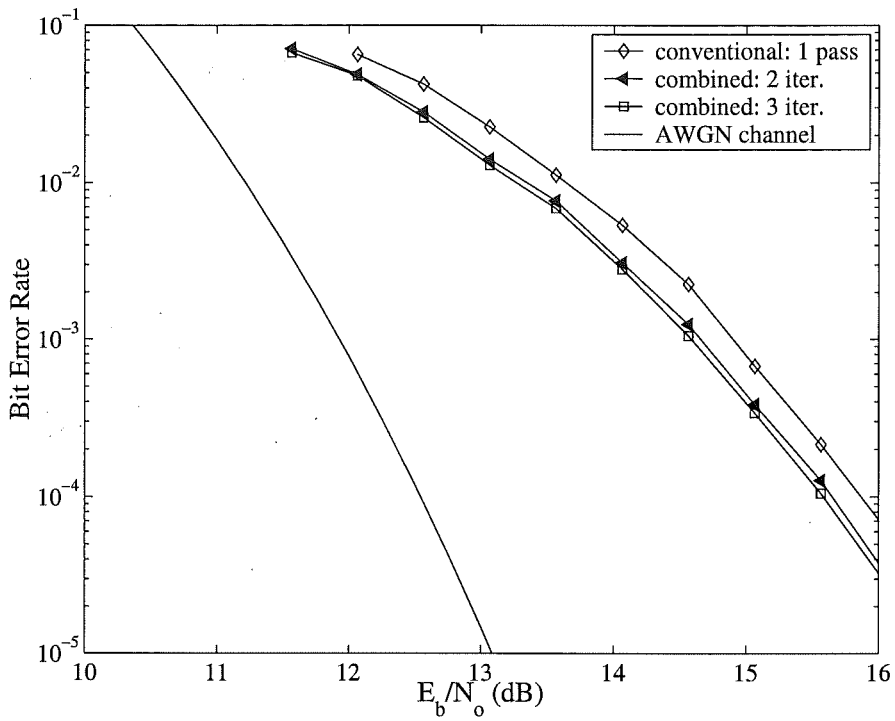


Figure 6.5: Performance of coded 64-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding system; and AWGN channel performance approximation [34]. Component codes: $C_1 = (16, 5, 8)$, $C_2 = (16, 11, 4)$, $C_3 = (16, 15, 2)$.

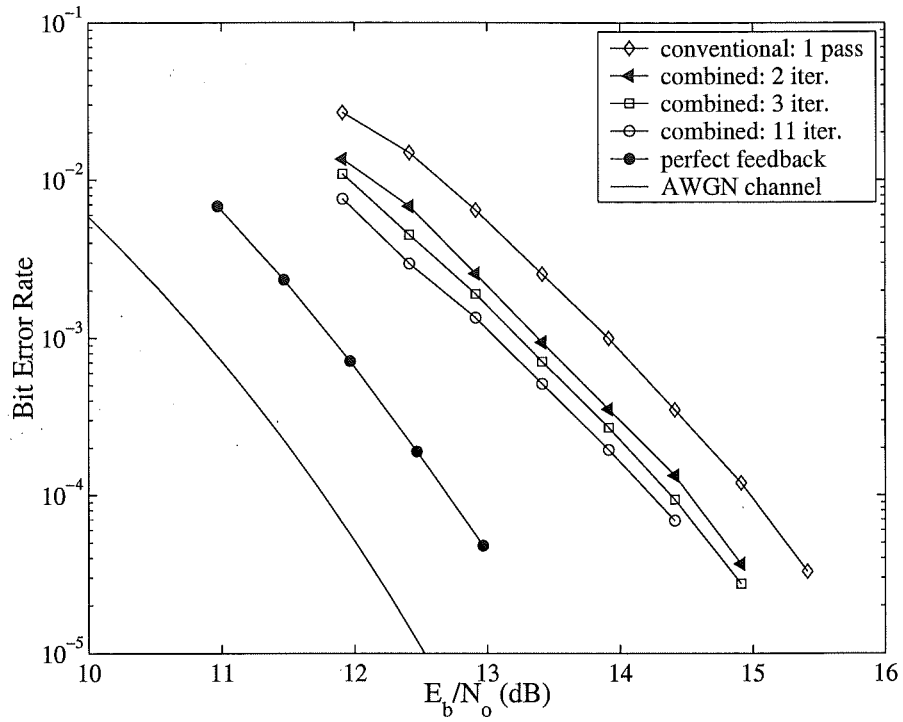


Figure 6.6: Performance of coded 64-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding; and AWGN channel performance approximation [34]. Component codes: $C_1 = (16, 1, 16)$, $C_2 = (16, 5, 8)$, $C_3 = (16, 11, 4)$.

mance between iterations of case three, compared to case two may be because its last non-trivial code has a small error coefficient of $A_{d_{min}} = 140$ – compared to the error coefficient of case two, $A_{d_{min}} = 2^{15}$. A large error coefficient restricts the overall performance in the multipath fading channel just as in the AWGN channel. There is also improved performance gain due to each iteration in case three, compared with earlier results using the code with a block length of 32 symbols. This may also partially be due to the fact that the last non-trivial code in case three, $C_3 = (16, 11, 4)$, has a small error coefficient [34] as well as to the identical feedback filter and code block lengths.

Figure 6.6 also shows the simulated performance when using conventional equalisation and decoding and assuming perfect feedback in the DFE. Here, the overall performance gain using conventional equalisation and decoding with perfect feedback rather than hard-decision decoded is approximately 2.3 dB. The gain using perfect feedback over combined equalisation and decoding with eleven iterations is about 1.5 dB. Therefore, by using the combined system we are again recovering approximately one third of the loss in performance from imperfect DFE feedback by using the combined structure with eleven iterations.

6.5 Alternative 96-QAM and 80-QAM Scenarios using Combined Equalisation and Decoding

As a comparison to the alternative 64-QAM cases of Section 6.4, we also illustrate the results of multilevel block encoding 96-QAM and 80-QAM

employing the two alternative sets of Reed-Muller component codes. Again, we employ the Reed-Muller component codes of cases two and three as given in Table 6.1. As will be seen, when employing these codes there is a performance gain as the number of iterations increase through the combined equalisation and decoding structure.

As illustrated in Figure 6.7 there is a performance gain between the first and second iteration and another marginal gain between the second and third iteration. This result has similar characteristics to the 64-QAM configuration illustrated in Figure 6.5. The simulations also gave similar results for 80-QAM employing the case two hierarchy of codes (the 80-QAM results are not displayed in this Section since their performance characteristics are so similar to 96-QAM). Figure 6.8 illustrates the performance of 96-QAM employing the hierarchy of component codes described as case three. At a BER of 10^{-4}, the performance gain between the first and second iterations is approximately 0.54 dB. Between the second and eleventh iteration there is a further 0.2-dB gain. As in the similarly coded 64-QAM case there is a large gain between the first and second iterations. Given a sufficient number of iterations, the overall gain over the first iteration may approach a gain of 1 dB. Figure 6.9 illustrates that the similarly coded 80-QAM constellation also has the same performance trends as 96-QAM.

In these 96-QAM and 80-QAM configurations, we may draw the same conclusions as in Section 6.4 – that is, the improved performance of case three compared to case two may be partially attributed to the much smaller error coefficient of the overall code. Using a code with a similar block length as

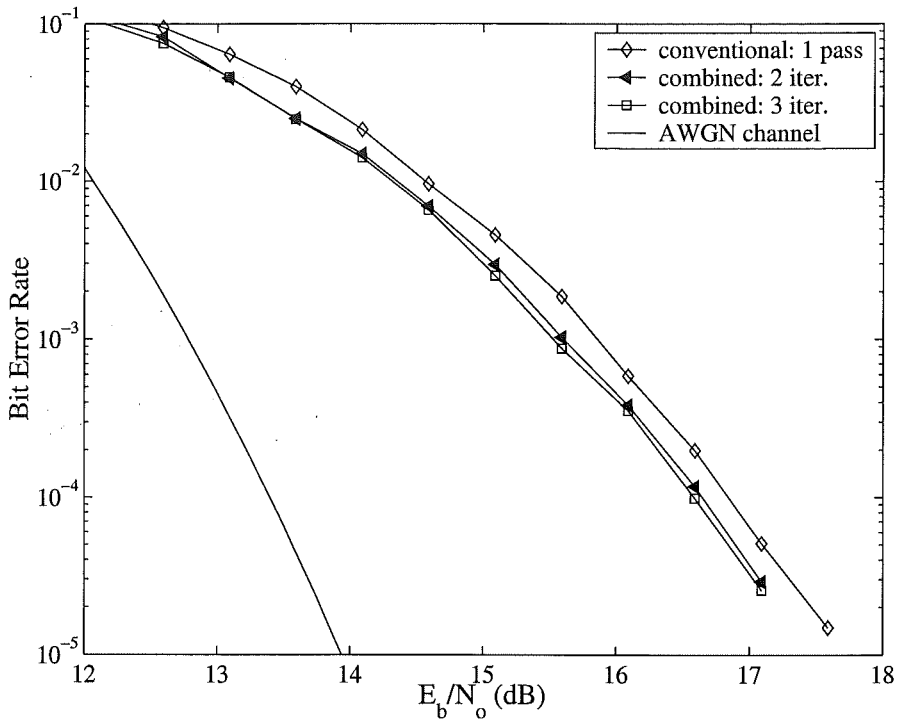


Figure 6.7: Performance of coded 96-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding; and AWGN channel performance approximation [34]. Component codes: $C_1 = (16, 5, 8)$, $C_2 = (16, 11, 4)$, $C_3 = (16, 15, 2)$.

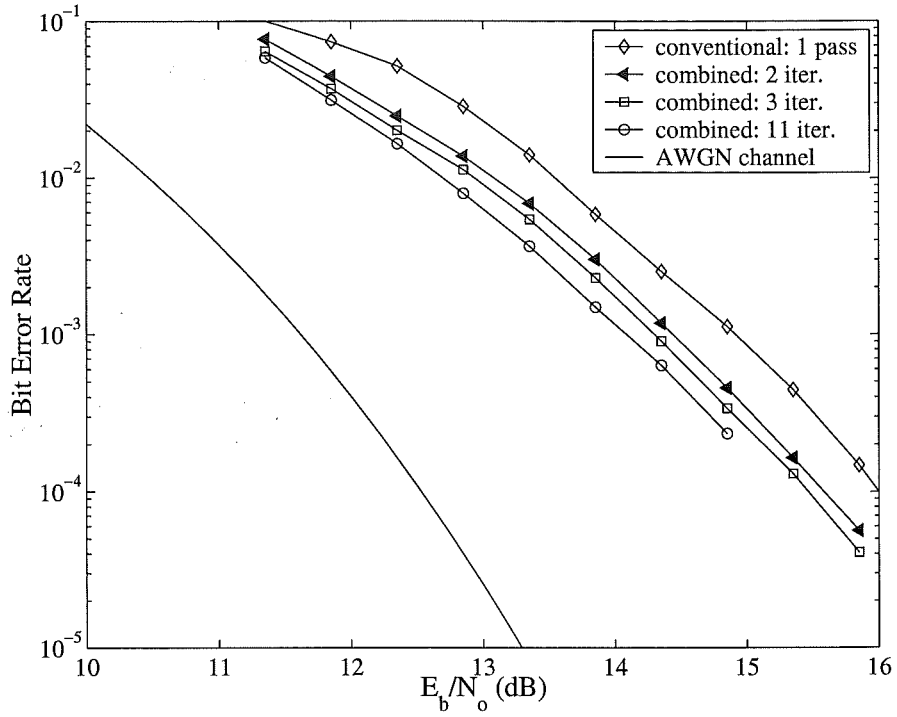


Figure 6.8: Performance of coded 96-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding system; and AWGN channel performance approximation [34]. Component codes: $C_1 = (16, 1, 16)$, $C_2 = (16, 5, 8)$, $C_3 = (16, 11, 4)$.

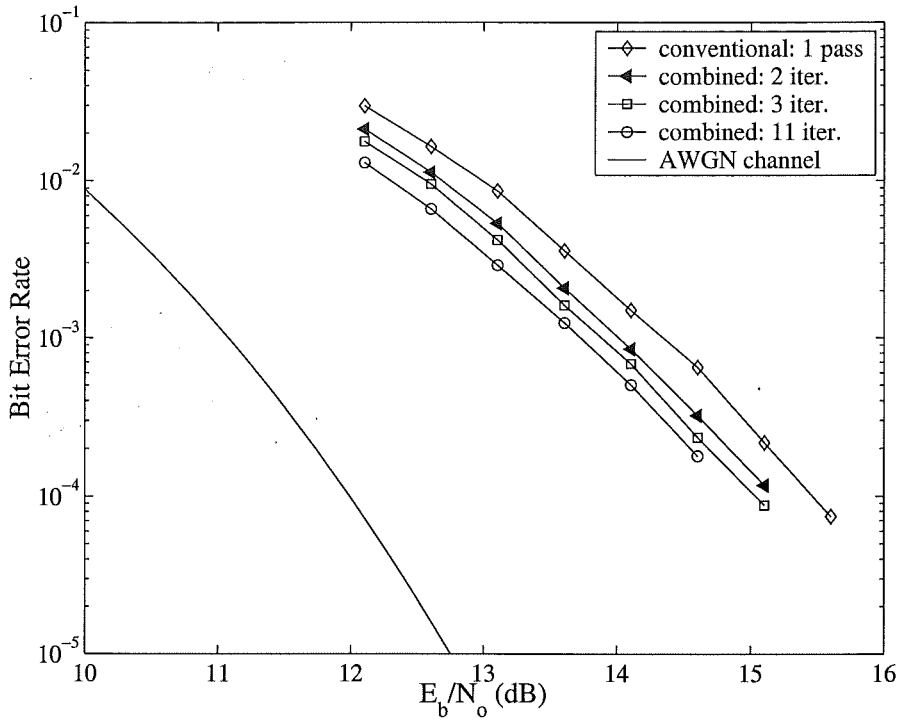


Figure 6.9: Performance of coded 80-QAM using conventional equalisation/soft-decision decoding; iterative combined equalisation/decoding system; and AWGN channel performance approximation [34]. Component codes: $C_1 = (16, 1, 16)$, $C_2 = (16, 5, 8)$, $C_3 = (16, 11, 4)$.

the feedback filter may also contribute to the overall performance improvement. Furthermore, we may also surmise that the shaping code has little or no effect on the iterative process. As we have seen in Figures 6.6, 6.8 and 6.9, the performance gains achieved via the unshaped or shaped constellation are approximately the same.

6.6 Performance of the Alternative System Models

All simulations discussed thus far have had both the forward filter and the feedback filter tap weights updated during each iteration. We now illustrate the performance of the more practical system configurations suggested in Equations (5.15) and (5.16) whereby the forward filter is not updated on every iteration.

Figure 6.10 illustrates performance when the forward filter is updated on only the first, or the first and second iterations. Again, we multilevel encode 64-QAM using the Reed-Muller component codes given as case one in Table 6.1. Here we have considered a total of five iterations. We observe that when we update the forward filter on only the first pass as in Equation (5.15), the performance is only very slightly worse than continuous updating. When we update the forward filter on only the first two iterations as in Equation (5.16), performance is essentially equivalent to updating it on every iteration.

We may, therefore, specify a practicable system whereby the forward filter is updated on the first pass only. Although there is a slight loss in performance using this model, it is not substantially worse than updating

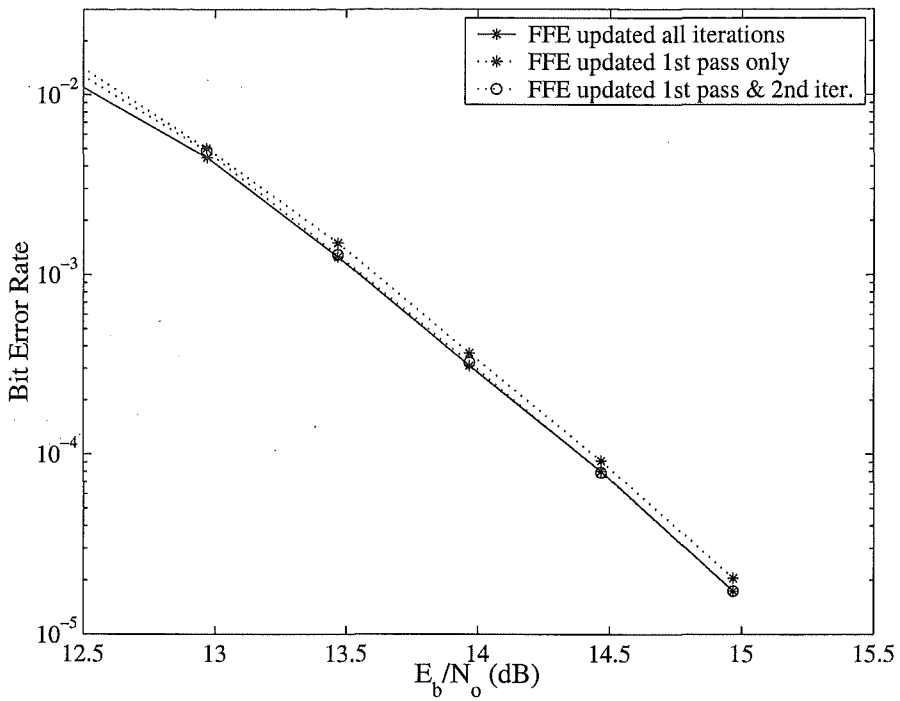


Figure 6.10: Coded 64-QAM – component codes: $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$, $C_3 = (32, 26, 4)$ – using combined equalisation/decoding implementing five iterations. Comparison of updating the forward filter during every iteration, versus updating it on only the first pass or only the first two iterations.

the forward filter every time. Furthermore, a typical iterative system has a large time delay between receiving the data and the final iteration. For many applications, for example voice communications, a large delay would be unacceptable.

It is well documented in [43] that in an optimal situation the forward and feedback filters may be operated separately. The simulation results displayed in Figure 6.10 indicate that this is true in practice even though filter optimality is not fully achieved during training. Therefore, to implement a practical system, as in Equation (5.15), the forward filter would receive, process and store s_i in real-time. The combined feedback filter and decoding would then be implemented as a separate entity from the forward filter. It would perform the iterative process and the recurring updating of the feedback filter using a separate processor. This processor would be operated at much higher speed than real-time, minimising the delay of successive iterations.

6.7 Soft-Decision Decoding Higher Level Codes

We now consider the effect of soft-decision decoding higher level component codes in the multilevel decoder. Using the 64-QAM constellation scenario, we illustrate two examples whereby we implement soft-decision (trellis) decoding on both C_1 and C_2. Thus the decoder outputs are closer to maximum likelihood and decoding is closer to optimal in a sequence sense.

The first example uses the hierarchy of component codes previously described as case one in Table 6.1. As a benchmark for this example, we also plot the performance of conventional equalisation and decoding as in Figure 6.2.

The second example uses the hierarchy of component codes described as case three given in Table 6.1. For this case, the benchmark results used are the performance of conventional equalisation and decoding as in Figure 6.6. In these examples, we show system performance using several iterations through the combined structure.

Figure 6.11 illustrates the 64-QAM example utilising the first case of component codes. Here, at a BER of 10^{-4}, the performance gain of the conventional system when soft-decision decoding both C_1 and C_2 is approximately 0.25 dB greater than when soft-decision decoding C_1 only. Similar to the results in Figure 6.2, when soft-decision decoding both C_1 and C_2 we also achieve a performance gain of around 0.25 dB at a BER of 10^{-4} between the first two iterations. After five iterations, the total performance gain at a BER of 10^{-4} is approximately 0.5 dB over the first iteration.

Figure 6.12 shows the performance using the third hierarchy of component codes. Here, at a BER of 10^{-4}, the performance gain of the conventional system when soft-decision decoding both C_1 and C_2 (as opposed to C_1 only) is approximately 0.55 dB. Again, coding gains for extra iterations are relative to the gain due to soft-decision decoding C_2. One may note that this gain is somewhat larger than the first example – this may be due partially to the similar feedback filter and code lengths. Note in Figure 6.12, since there is a performance gain when soft-decision decoding both C_1 and C_2, the iterative performance is closer to perfect feedback than it is in Figure 6.2.

In both the examples shown (and others simulated, but not shown), soft-decision decoding higher level codes will provide improvement in overall

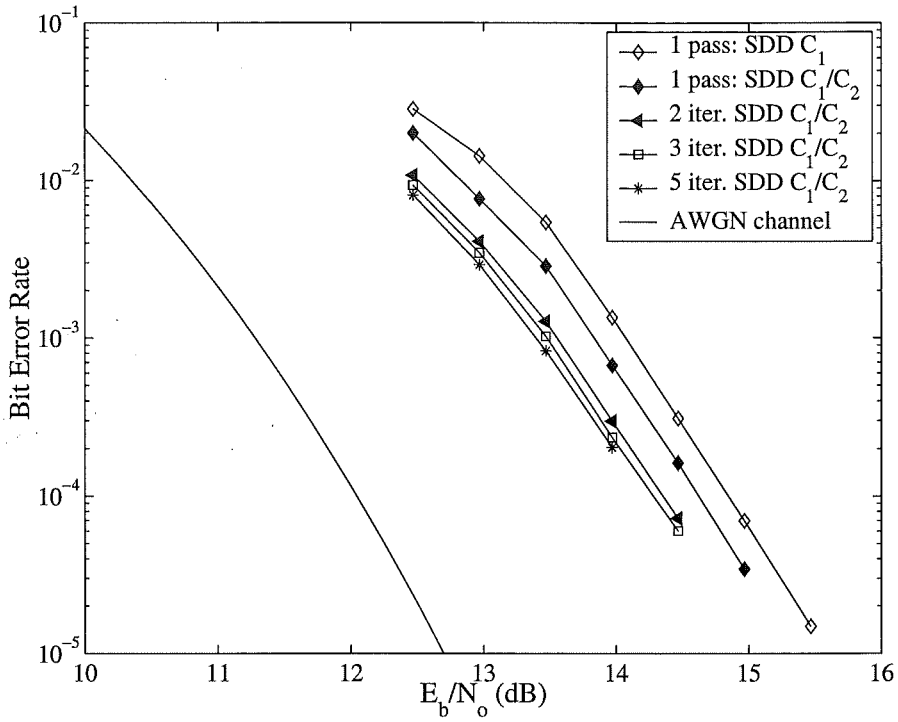


Figure 6.11: Comparison of soft-decision decoding C_1 using conventional equalisation/decoding versus soft-decision decoding both C_1 and C_2 for one, two, three and five iterations using the combined system. Constellation: 64-QAM; component codes: $C_1 = (32, 6, 16)$, $C_2 = (32, 16, 8)$, $C_3 = (32, 26, 4)$.

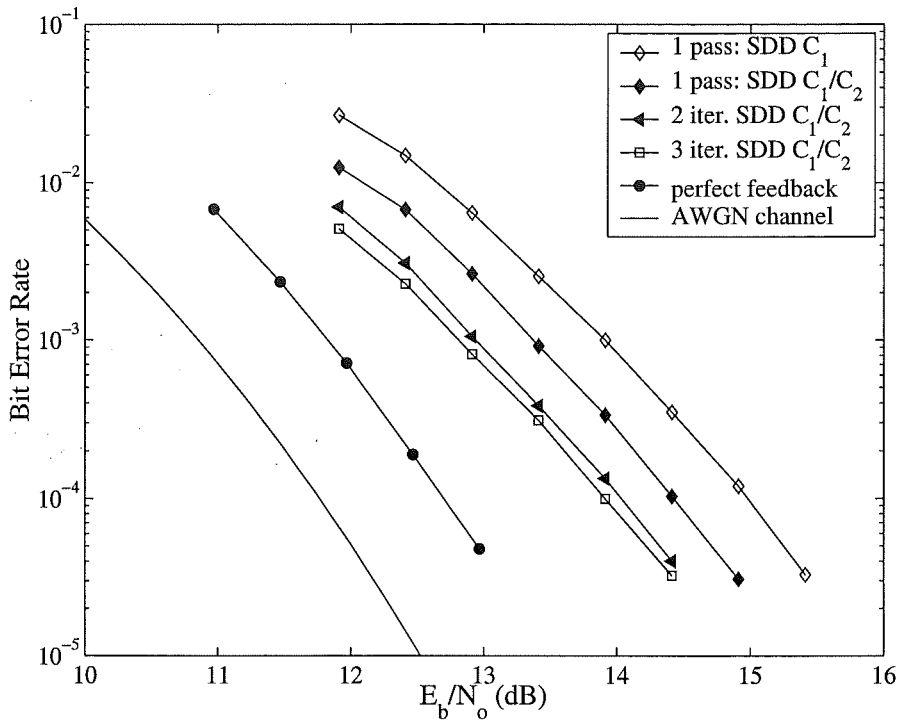


Figure 6.12: Comparison of soft-decision decoding C_1 using conventional equalisation/decoding versus soft-decision decoding both C_1 and C_2 for one, two and three iterations using the combined system. Perfect feedback when Soft-decision decoding C_1 and C_2 is also shown. Constellation: 64-QAM; component codes: $C_1 = (16, 1, 16)$, $C_2 = (16, 5, 8)$, $C_3 = (16, 11, 4)$.

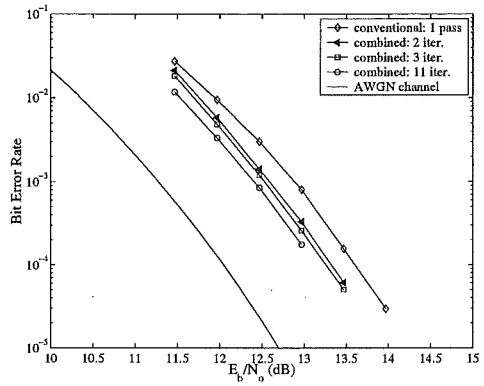
equalisation/decoding performance. This improvement is expected since as we soft-decision decoded more higher level component codes, the overall decoder performance becomes closer to maximum likelihood. These extra gains are obtained at the expense of decoding delay and complexity. However, by using a set of carefully chosen component codes, soft-decision decoding C_1 and C_2 may achieve a large overall gain with minimal decoding complexity.

6.8 Performance using Alternative Channel Coefficients

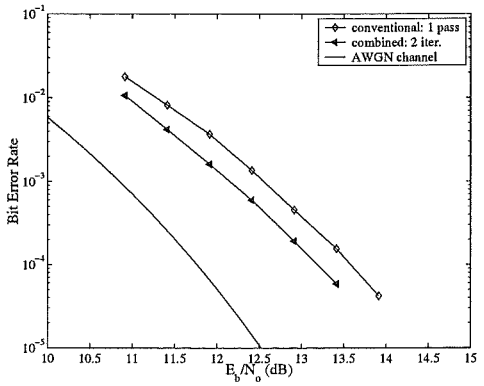
Finally, we show some results using alternative channel coefficients. We vary the notch depth and position used in all the previous examples. The examples given are for the hierarchy of component codes given as case one and case three in Table 6.1.

Figures 6.13(a) and 6.13(b) show the performance using the hierarchy of component codes described as case one and case three respectively. These Figures illustrate the performance when using multipath channel coefficients such that the path amplitudes are $c_1 = \sqrt{0.7}$, $c_2 = \sqrt{0.3}e^{(-j\pi/1.5)}$ and the relative delay of the second path is set to $\tau_2 = \frac{1}{3}T$. These channel coefficients equate to an 8-dB notch near band edge. Essentially, this channel is the same channel as all previous examples but with a smaller notch depth. As may be seen, when performing extra iterations in the combined system, a performance gain is obtained as in all other examples discussed. The overall performance has improved from the the system in Figure 6.2 due to the smaller notch depth.

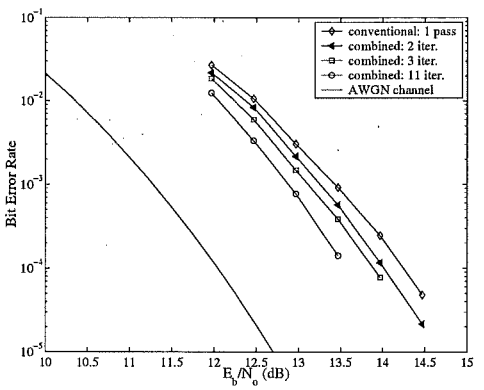
Figures 6.13(c) and 6.13(d) also illustrate the performance using the hierarchy of component codes described as case one and case three respectively.



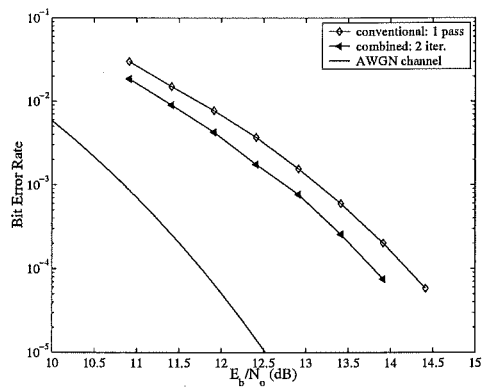
(a) channel one – case one



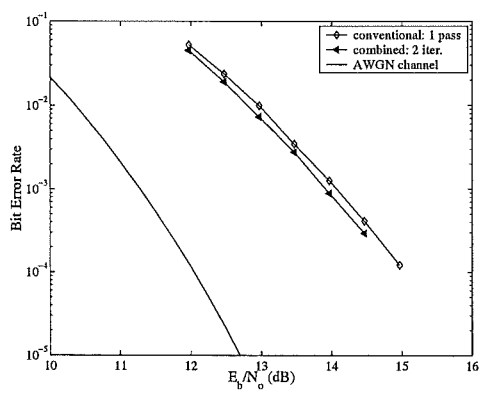
(b) channel one – case three



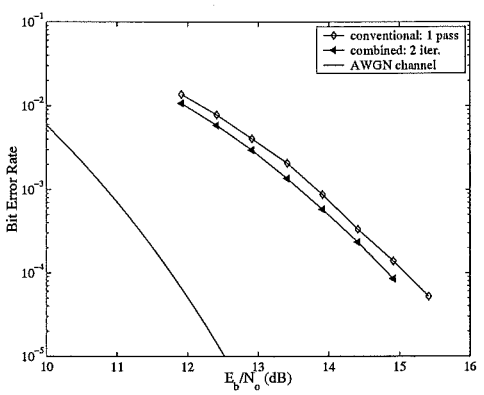
(c) channel two – case one



(d) channel two – case three



(e) channel three – case one



(f) channel three – case three

Figure 6.13: Performance of coded 64-QAM using conventional equalisation and soft-decision decoding; the iterative combined equalisation/decoding system; and the performance approximation on an AWGN channel [34].

However, these Figures illustrate the performance when using a multipath channel with the path amplitudes $c_1 = \sqrt{0.7}$, $c_2 = \sqrt{0.3}e^{(j\pi/0.9)}$ and the relative delay of the second path is set to $\tau_2 = \frac{1}{3}T$. These channel coefficients also equate to an 8-dB notch – however, this notch is located near band centre. Again, extra iterations using the combined system give performance gain.

Figures 6.13(e) and 6.13(f) also illustrate the performance using the hierarchy of component codes described as case one and case three respectively. These Figures illustrate the performance when using a multipath channel with the path amplitudes $c_1 = \sqrt{0.7}$, $c_2 = -\sqrt{0.3}$ and the relative delay of the second path is set to $\tau_2 = \frac{1}{2}T$. These channel coefficients also equate to an 8-dB notch and this notch is located at band centre. Again, extra iterations using the combined system give performance gain.

As may be seen from the figures in Figure 6.13, when the notch is near or at band centre, the performance is slightly worse than when the notch is at band edge – as would be expected.

6.9 Summary

We have simulated the performance of the iterative combined structure using 64-QAM, 80-QAM and 96-QAM constellations with associated shaping codes and various component code hierarchies. In the examples shown, by making a second iteration in this way and feeding back decisions from the decoder we are able to obtain at least 0.25-dB performance gain over one pass through a conventional equaliser and soft-decision decoder. Further iterations also show performance gains. However, the largest incremental performance

gain is between the first pass and second iteration with subsequent iterations yielding smaller gains each time. We have also shown the performance when using channels with alternative channel coefficients. Other channels were also simulated but have not been reported here. All the results show similar performance characteristics to the results from the original system example using the 25 dB notch near band edge.

We also reviewed the performance of 64-QAM, 80-QAM and 96-QAM employing a code block length equal to the DFE feedback filter length. Providing the component codes are carefully chosen, given around 10 iterations the overall performance gain over the conventional system tends towards 1 dB. If the hierarchy of component codes has a large error coefficient limiting the potential performance, improvement is limited. As an example, the case two hierarchy of component codes has a large overall error coefficient. In each case, we achieve a gain of around 0.25 dB between the first and second iteration and then marginal gains for any extra iterations. This illustrates that an iterative combined system with properly chosen component codes can produce a reasonably large gain.

Also observed is the performance gain when soft-decision decoding both C_1 and C_2 – as opposed to C_1 only. At a BER of 10^{-4}, an absolute performance gain (independent of iterations) of approximately 0.25 dB is achieved. This improvement is expected – since as we soft-decision decoded more higher level component codes, the overall decoder performance becomes closer to maximum likelihood. Although soft-decision decoding higher level codes will provide some improvement in overall equalisation/decoding performance, it is obtained

at the expense of decoding delay and complexity.

Chapter 7

CONCLUSIONS

7.1 Introduction

The research surrounding digital communication, coded modulation and equalisation is an interesting and important field. It will continue to be so with the development in communication technology and the growing demand for high-speed, low cost and efficient communications systems. There are many possible areas of research within this broad field.

This Thesis has made original research contributions in two areas within the field of digital communications. The first is based on a coded modulation scheme, referred to as multilevel coding – which is an alternative to TCM [31]. Here, we have developed simple block code designs, have examined alternative decoding approaches and have developed an approximation to the error performance of multilevel block codes. The second is to describe a simple approach to combined decoding and equalisation of multilevel block codes. Its performance is compared to the performance of conventional, concatenated

equalisation and decoding. We also discuss a simplified combined equalisation and decoding structure for practical implementation.

This Chapter concludes the research reported in this Thesis and gives some suggestions for ways in which this research may be continued.

7.2 Thesis Conclusions

We have considered multilevel coded signalling employing spectrally efficient modulation schemes. Multilevel codes are constructed using a hierarchy of component codes which are mapped to an expanded signal set [31]. They offer significant coding gain with modest decoding complexity [31].

We have developed an approximation to the error performance of multilevel block codes on the AWGN channel. The multilevel code examples used employ a hierarchy of three Reed-Muller codes; however, multilevel codes are not restricted to Reed-Muller codes. The modulation schemes are based on large two-dimensional or multidimensional QAM constellations. In each example shown, the analytical performance approximation is very close to the simulated results. The one anomaly occurs when using smaller constellations – for example, four-dimensional 24-QAM – where set partitioning with increasing MSED between signal points in each subset cannot be achieved at higher partition levels. However, even in these cases, the analytical result reasonably closely approximates the simulated performance.

We have also explored the trade-off between hard-decision decoding and soft-decision decoding each component code. It is assumed that the component codes are designed such that high level codes have a decreasing minimum

Hamming distance and the set partitioning is such that there is an increasing MSED between signal points. Therefore, soft-decision decoding the first component code gives a relatively large performance gain with little decoding effort given the right choice of component codes. Soft-decision decoding the second component code also gives a reasonably significant increase in performance, but with a high decoding complexity. We have concluded that as we soft decision decode higher level codes, an incremental but decreasing performance gain will be achieved. This is due to the fact that symbol labels in the higher level coded partitions are more dependent on the large MSED between nearest signal points (rather than the code's Hamming distance) to obtain equal performance.

Finally with regard to multilevel coding, the analytical results indicate that the error coefficient significantly influences performance, and therefore, should be considered during system design. In the present instance, this has meant allowing the parallel transitions of C_4 to dominate the exponential behaviour of the error-rate curve in order to obtain a reduced error coefficient.

The second area investigated combined equalisation and decoding. We have described an approach to combined decision-feedback equalisation and multilevel block decoding. The multilevel code uses a hierarchy of Reed-Muller component codes and large QAM constellations as described above.

This approach to combined equalisation and decoding is based on simple iterative decoding principles. In this structure, the symbols from the output of the decoder are re-encoded and re-generated into symbol points in the (transmitted) constellation, then used as the feedback portion of the DFE to

re-process the received symbol. This operation may be iterated as many times as desired. It is based on the fact that if we minimise the number of errors fed back into the DFE – that is, each iteration decreases the number of errors that the quantiser output would have otherwise fed back – the overall system performance improves. We have also shown that the proposed structure may be simplified for practical implementation by separating the forward filter from the combined feedback and equalisation process.

As examples, we have simulated the performance of the iterative combined structure using 64-QAM, 80-QAM and 96-QAM constellations with associated shaping codes and various component code hierarchies. The hierarchies of component codes employed either have the same or larger code block lengths than the feedback filter in the DFE. In all the examples shown, when making a second iteration by feeding back remodulated decisions from the decoder into the feedback filter, we obtain a performance gain over one pass through a conventional equaliser and soft-decision decoder. Further iterations also show performance gains. However, the largest incremental performance gain is between the first pass and second iteration with subsequent iterations yielding smaller gains each time. When using a component code block length of the same size as the feedback filter length, we obtain a slightly larger performance gain than when using a larger component code block length – provided that the performance is not hindered by a large error coefficient of the last non-trivial code.

As with the AWGN channel analysis, if the hierarchy of component codes has a large error coefficient, the potential performance improvement is

limited. We have illustrated that an iterative combined system with properly chosen component codes can produce a reasonably large performance gain over conventional, concatenated equalisation and decoding.

Finally, we have noted that when increasing the number of component codes that are soft-decision decoded, we obtain an absolute performance gain that is independent of any iterative processing. As we soft-decision decode more higher level component codes, the overall decoder performance becomes closer to maximum likelihood. It must be noted that soft-decision decoding higher level codes does provide some improvement in overall equalisation and decoding performance – however, it is obtained at the expense of decoding delay and decoder complexity.

7.3 Suggestions for Future Research

There are many areas in which this research may be extended. Particularly exciting possibilities are in the area of combined equalisation and decoding.

In terms of the multilevel block code research, an investigation into larger block length codes and alternative decoding methods are interesting areas for research. It would be an invaluable exercise to further investigate iterative decoding (Turbo-type decoding) of multilevel codes, similar to the work of [92]. The goal of this research would be to preserve a short decoding delay and retain the minimal decoding complexity of the multilevel codes while improving overall performance.

One may also investigate the use of codes with a longer block length and

higher code rate to raise the throughput of the overall multilevel code. When using longer block length codes, it may be possible to partially decode the lower level component codes and pass this partially decoded (soft) information to higher level codes. This may help to increase the speed of decoding large block length codes and decoding results may also improve. Further, this may be extended to iteratively decoding multilevel codes, whereby the soft information is improved at each iteration.

Since the focus of this Thesis was to achieve multilevel block codes with a short decoding delay, this should also remain the focus for iterative decoding methods and codes with longer block lengths.

In terms of continued research in combined equalisation and decoding, there are many interesting avenues to pursue. One obvious improvement to the suggested combined equalisation and decoding structure is to replace the hard quantiser of the DFE with a soft, MMSE estimator – cf. Equation (5.14). This would increase the performance of the first iteration of the combined system – thereafter, given further iterations, the overall performance of the combined system would also improve.

Another avenue to investigate is to incorporate iterative decoding (or feedback) of each component code in a multilevel code, combined with the iterative combined equalisation and decoding structure suggested in this Thesis. This may, again, improve overall system performance. The feedback in the multilevel code may be in the form of turbo-type decoding, whereby each component code is given some reliability information for use in latter iterations. Turbo-type decoding has been shown in various research initiatives to

improve overall system performance in coding schemes. Ideally, we wish to emulate perfect feedback in the DFE and the maximum-likelihood decoding of multilevel codes.

There is also scope for keeping one forward filter in the DFE and incorporating several feedback filters into the combined structure in a similar manner to [41] and/or [40]. Here, instead of each iteration through the feedback filter being a one path approximation to the Viterbi algorithm, more paths will be available to improve the reliability of each Viterbi approximation. At the end of the overall iterative process one may discard all the least reliable feedback filter contents and continue the process with the most reliable output.

Further, other channels (both more severe and more benign) may be used in the simulations to further document the channel performance. One may also wish to investigate the exact change (improvement) of the dispersive fade margin when using the combined structure as opposed to conventional concatenation of equalisation and decoding.

Essentially, any future research should focus particularly at improving the reliability of the data fed back into the equaliser. In the channels of interest, the equaliser feedback is one of the main limiting factors in increasing overall performance.

BIBLIOGRAPHY

[1] R. AARON, M. DECINA, AND R. SKILLEN, "Electronic Commerce: Enablers and Implications", *IEEE Communications Magazine*, vol. 37, no. 9, pp. 47–52, Sept. 1999.

[2] C. L. HOEWING, "Moving to Competition: New Zealand's Innovative Regulatory Model for the Telecommunications Market Place", *IEEE Communications Magazine*, vol. 37, no. 10, pp. 110–113, Oct. 1999.

[3] D.P. TAYLOR AND P. R. HARTMAN, "Telecommunications by Microwave Digital Radio", *IEEE Communications Magazine*, vol. 24, no. 8, pp. 11–16, Aug. 1986.

[4] L.J. GREENSTEIN AND M. SHAFI, Eds., *Microwave Digital Radio*, IEEE Press, 1988.

[5] P. LETTIERI AND M.B SRIVASTAVA, "Advances in Wireless Terminals", *IEEE Personal Communications*, vol. 60, no. 1, pp. 6–19, Feb. 1999.

[6] A. DUTTA-ROY, "Fixed Wireless Routes for Internet Access", *IEEE Spectrum*, vol. 36, no. 9, pp. 61–69, Sept. 1999.

[7] M. ZENG, A. ANNAMALAI, AND V.K. BHARGAVA, "Recent Advances in Cellular Wireless Communications", *IEEE Communications Magazine*, vol. 37, no. 9, pp. 128–138, Sept. 1999.

[8] V.K. GARG AND E.L. SNEED, "Digital Wireless Local Loop System", *IEEE Communications Magazine*, vol. 34, no. 10, pp. 112–115, Oct. 1996.

[9] R. WESTERVELD AND R. PRASAD, "Rural Communications in India Using Fixed Cellular Radio Systems", *IEEE Communications Magazine*, vol. 32, no. 10, pp. 70–77, Oct. 1994.

[10] M.H. CALLENDAR, "Future Public Land Mobile Telecommunications Systems", *IEEE Personal Communications Magazine*, pp. 18–22, Fourth Quarter 1994.

[11] S. BENEDETTO, E. BIGLIERI, AND V. CASTELLANI, *Digital Transmission Theory*, Prentice-Hall, Inc., 1987.

[12] C.E. SHANNON, "A Mathematical Theory of Communications", *Bell Systems Technical Journal*, vol. 27, pp. 379–423, 1948.

[13] J.G. PROAKIS, *Digital Communications*, McGraw-Hill, second edition, 1989.

[14] R.E. ZIEMER AND W.H. TRANTER, *Principles of communications: Systems, Modulation and Noise*, John Wiley & Sons, Inc., fourth edition, 1995.

[15] W. GAPPMAIR, "Claude E. Shannon: The 50th Anniversary of Information Theory", *IEEE Communications Magazine*, vol. 4, no. 37, pp. 102–105, Apr. 1999.

[16] S.B. WICKER AND V.K. BHARGAVA, Eds., *Reed-Solomon Codes and their Applications*, IEEE Press, 1994.

[17] R.W. HAMMING, "Error Detecting and Correcting Codes", *Bell Systems Technical Journal*, vol. 29, pp. 147–160, 1950.

[18] M.J.E. GOLAY, "Notes on Digital Coding", *Proceedings of the IEEE*, vol. 37, pp. 657, 1949.

[19] T.M. THOMPSON, *From Error-Correcting Codes Through Sphere Packings to Simple Groups*, Number 21 in The Carus Mathematical Monographs. The Mathematical Association of America, 1983.

[20] E.R. BERLEKAMP, Ed., *Key Papers in the Development of Coding Theory*, IEEE Press, 1974.

[21] I.S. REED, "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme", *IEEE Transactions on Information Theory*, vol. 4, pp. 38–49, 1954.

[22] D.E. MULLER, "Applications of Boolean Algebra to Switching Circuit Design and to Error Detection", *IEEE Transactions on Computers*, vol. 3, pp. 6–12, 1954.

[23] S.B. WICKER, *Error Control Systems for Digital Communications and Storage*, Prentice-Hall, 1995.

[24] P. ELIAS, "Coding for Noisy Channels", *IRE Nat. Conv. Rec., pt. 4*, pp. 37–46, 1955.

[25] G.D. FORNEY JR., "Concatenated Codes", *Cambridge Mass.: MIT Press*, 1966.

[26] A.J. VITERBI, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, Apr. 1967.

[27] H. IMAI AND S. HIRAKAWA, "A New Multilevel Coding Method using Error-Correcting Codes", *IEEE Transactions on Information Theory*, vol. IT-23, no. 3, pp. 371–377, May 1977.

[28] G. UNGERBOECK, "Channel Coding with Multilevel/Phase Signals", *IEEE Transactions on Information Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.

[29] C. BERROU, A. GLAVIEUX, AND P. THITIMAJSHIMA, "Near Shannon Limit Error-Correcting Coding: Turbo Codes", in *Proceedings of ICC'93, Geneva, Switzerland*, May 1993, pp. 1064–1070.

[30] J.H. CONWAY AND N.J.A. SLOANE, *Sphere Packings, Lattices and Groups*, Springer-Verlag New York, 1988.

[31] G.J. POTTIE AND D.P. TAYLOR, "Multilevel Codes Based on Partitioning", *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 87–98, Jan. 1989.

[32] A.R. CALDERBANK, "Multilevel Codes and Multistage Decoding", *IEEE Transactions on Communications*, vol. 37, no. 3, pp. 222–229, Mar. 1989.

[33] D.P. TAYLOR AND W.L. SO, "On the Design and Performance of Simple Multi-Level Codes", in *5th Tirrenia International Workshop on Digital Communications, Italy*, M. LUISE E. BIGLIERI, Ed., 1992, pp. 213–224.

[34] K.O. HOLDSWORTH, D.P. TAYLOR, AND R.T PULLMAN, "On the Error Performance of Multilevel Block Coded Modulation", *IEEE Communications Letters*, vol. 3, no. 11, pp. 311–313, Nov. 1999.

[35] K.O. HOLDSWORTH, D.P. TAYLOR, AND R.T. PULLMAN, "On the Error Performance of Multilevel Block Coded Modulation", *ISWC'99 IEEE International Symposium on Wireless Communications*, pp. 21–22, June 1999.

[36] E.A. LEE AND D.G. MESSERSCHMITT, *Digital Communication*, Kluwer Academic Publishers, Boston, 1988.

[37] S.U.H. QURESHI, "Adaptive Equalization", *Proceedings of the IEEE*, vol. 73, no. 9, pp. 1349–1385, Sept. 1985.

[38] J. SALZ, "Optimum Mean-Square Decision Feedback Equalization", *The Bell Systems Technical Journal*, vol. 52, no. 8, pp. 1341–1373, Oct. 1973.

[39] G.-K. LEE, S.B. GELFAND, AND M.P. FITZ, "Bayesian Techniques for Blind Deconvolution", *IEEE Transactions on Communications*, 1993.

[40] S.L. ARIYAVISITAKUL AND Y. (G.) LI, "Joint Coding and Decision Feedback Equalization for Broadband Wireless Channels", *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, pp. 1670–1678, Dec. 1998.

[41] D. YELLIN, A. VARDY, AND O. AMRANI, "Joint Equalization and Coding for Intersymbol Interference Channels", *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 409–425, Mar. 1997.

[42] C. HEEGARD AND S.B. WICKER, *Turbo Coding*, Kluwer Academic Publishers, 1999.

[43] C.A. BELFIORE AND J.H. PARK JR., "Decision Feedback Equalization", *Proceedings of the IEEE*, vol. 67, no. 8, pp. 1143–1157, Aug. 1979.

[44] L.-F. WEI, "Trellis-Coded Modulation with Multidimensional Constellations", *IEEE Transactions on Information Theory*, vol. IT-33, no. 4, pp. 483–501, Jan. 1987.

[45] F.J. MACWILLIAMS AND N.J.A. SLOANE, *The Theory of Error-Correcting Codes*, vol. 16, Elsevier Science B.V., 1977.

[46] G.D. FORNEY AND L.-F. WEI, "Multidimensional Constellations - Part I: Introduction, Figures of Merit, and Generalized Cross Constellations", *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 877–892, Aug. 1989.

[47] K.O. HOLDSWORTH, D.P. TAYLOR, AND R.T. PULLMAN, "On Combined Equalization and Decoding of Multilevel Coded Modulation", *ISWC'99 IEEE International Conference on Communications*, , no. 31.8, June 2000.

[48] W.D. RUMMLER, R.P. COUTTS, AND M. LINIGER, "Multipath Fading Channel Models for Microwave Digital Radio", *IEEE Communications Magazine*, vol. 24, no. 11, pp. 30–42, Nov. 1986.

[49] M.P. FITZ AND J.P. SEYMOUR, "On the Bit Error Probability of QAM Modulation", *International Journal of Wireless Information Networks*, vol. 1, no. 2, 1994.

[50] S. HAYKIN, *Communications Systems*, John Wiley and Sons, Inc., third edition, 1994.

[51] T.S. RAPPAPORT, *Wireless Communications: Principles and Practice*, Prentice Hall, Inc., 1996.

[52] W.T. WEBB AND L. HANZO, *Modern Quadrature Amplitude Modulation: Principles and Applications for Fixed and Wireless Communications*, IEEE Press, 1994.

[53] R.D. GITLIN, J.F. HAYES, AND S.B. WEINSTEIN, *Data Communications Principles*, Plenum Press, 1992.

[54] J.M. WOZENCRAFT AND I.M. JACOBS, *Principles of Communications Engineering*, John Wiley and Sons, Inc., 1965.

[55] B. SKLAR, *Digital Communications: Fundamentals and Applications*, Prentice-Hall International, Inc., 1988.

[56] J.G. PROAKIS AND M. SALEHI, *Communication Systems Engineering*, Prentice-Hall, Inc., 1994.

[57] B. HART, *MLSE Diversity Receiver Structures*, PhD thesis, University of Canterbury, Christchurch, New Zealand, 1996.

[58] W.H. TRANTER AND K.L. KOSBAR, "Simulation of Communication Systems", *IEEE Communications Magazine*, July 1994.

[59] M. SHAFI, "Statistical Analysis/Simulation of a Three Ray Model for Multipath Fading with Applications to Outage Prediction", *Technical Report, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada*, July 1986.

[60] J. DOBLE, *Introduction to Radio Propagation for Fixed and Mobile Communications*, Artech House Publishers, 1996.

[61] D.P. TAYLOR, G.M. VITETTA, B.D. HART, AND A. MAMMELA, "Wireless Channel Equalization", *ETT*, vol. 9, no. 2, pp. 1–27, March-April 1988.

[62] W.D. RUMMLER, "A Multipath Channel Model for Line-of-Sight Digital Radio Systems", *ICC'78 Conference Proceedings*, 1978.

[63] W.D. RUMMLER, "A New Selective Fading Model: Application to Propagation Data", *Bell Systems Technical Journal*, vol. 58, pp. 1037–1071, may-Jun. 1979.

[64] S. HAYKIN, *Adaptive Filter Theory*, Prentice-Hall, Inc., third edition, 1996.

[65] J.K. CHAMBERLAIN, F.M. CLAYTON, H. SARI, AND P. VANDAMME, "Receiver Techniques for Microwave Digital Radio", *IEEE Communications Magazine*, vol. 24, no. 11, pp. 43–54, Nov. 1986.

[66] C. SCHLEGEL, *Trellis Coding*, IEEE Press, 1997.

[67] E. AYANOGLU, K.Y. ENG, AND M.J. KAROL, "Wireless ATM: Challenges and Proposals", *IEEE Personal Communications Magazine*, vol. 3, no. 4, pp. 18–34, Aug. 1996.

[68] A.J. VITERBI, "Convolutional Codes and Their Performance in Communications Systems", *IEEE Transactions on Communications Technology*, vol. COM-19, no. 5, pp. 751–772, Oct. 1971.

[69] B HONARY AND G. MARKARIAN, *Trellis Decoding of Block Codes: A Practical Approach*, Kluwer Academic Publishers, 1997.

[70] M.V. EYUBOGLU AND S.U.H. QURESHI, "Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback", *IEEE Transactions on Communications*, vol. 36, no. 1, pp. 13–20, Jan. 1988.

[71] R.A. SILVERMAN AND M. BALSER, "Coding for Constant-Data Rate Systems", *IRE Transactions on Information Theory*, pp. 50–63, 1954.

[72] D. CHASE, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information", *IEEE Transactions on Information Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.

[73] K.S. SHANMUGAM, *Digital and Analog Communication Systems*, John Wiley and Sons, Inc., 1979.

[74] W.T. FIRMANTO AND T.A. GULLIVER, "Code Combining of Reed-Muller Codes in an Indoor Wireless Environment", *To appear in Wireless Personal Communications*, 1999.

[75] G.D. FORNEY, "The Viterbi Algorithm", *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[76] G. UNGERBOECK, "Trellis-Coded Modulation with Redundant Signal Sets Part I: Introduction", *IEEE Communications Magazine*, vol. 25, no. 3, pp. 5–21, Feb. 1987.

[77] G.D. FORNEY, R.G. GALLAGER, G.R. LANG, AND S.U. QURESHI, "Efficient Modulation for Band-Limited Channels", *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, no. 5, pp. 632–647, Sept. 1984.

[78] A.R. CALDERBANK AND L.H. OZAROW, "Nonequiprobable Signaling on the Gaussian Channel", *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 726–740, July 1990.

[79] E.L. CUSACK, "Error Control Coding for QAM Signalling", *Electronic Letters*, vol. 20, no. 2, pp. 62–63, Jan. 1984.

[80] S.I. SAYEGH, "A Class of Optimum Block Codes in Signal Space", *IEEE Transactions on Communications*, , no. 10, pp. 1043–1045, Oct. 1986.

[81] V.V. GINZBURG, "Multidimensional Signals for a Continuous Channel", *Problemy Peredachi Informatsii*, vol. 20, no. 1, pp. 28–46, January-March 1984, translated from Russian by Plenum Publishing Corporation, 1984.

[82] R.M. TANNER, "Algebraic Construction of Large Euclidean Distance Combined Coding/Modulation Systems", Tech. Rep., The University of California, Santa Cruz: UCSC-CRL-87-7, June 1987.

[83] T. KASAMI, T. TAKATA, T. FUJIWARA, AND S. LIN, "On Multilevel Block Modulation Codes", *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 965–975, July 1991.

[84] T. TAKATA, S. UJITA, T. KASAMI, AND S. LIN, "Multistage Decoding of Multilevel Block M-PSK Modulation Codes and its Performance Analysis", *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1204–1218, July 1993.

[85] L. ZHANG AND B. VUCETIC, "New MPSK BCM codes for Rayleigh Fading Channels", *Singapore ICCS/ISITA '92. 'Communications on the Move'*, vol. 3, pp. 857–861, 1992.

[86] L. ZHANG AND B. VUCETIC, "Multilevel Block Codes for Rayleigh-Fading Channels", *IEEE Transactions on Communications*, vol. 43, no. 1, pp. 24–31, Jan. 1995.

[87] Y.-M. GU AND C. W. LEE, "Multilevel Coded 16-QAM Schemes for Rayleigh Fading Channels", *Electronic letters*, vol. 33, no. 4, pp. 265–267, Feb. 1997.

[88] R. VAN NOBELEN AND D.P. TAYLOR, "Multiple Symbol Differentially Detected Multilevel Codes for the Rayleigh-Fading Channel", *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1529–1537, Dec. 1997.

[89] H. HERZBERG, Y. BE'ERY, AND J. SNYDERS, "Concatenated Multilevel Block Coded Modulation", *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 41–49, Jan. 1993.

[90] J. WU AND D.J. COSTELLO JR, "New Multilevel Codes over GF(q)", *IEEE Transactions on Information Theory*, vol. 38, no. 3, pp. 933–939, May 1992.

[91] M. ISAKA, R.H. MORELOS-ZARAGOZA, M.P.C. FOSSORIER, S. LIN, AND H. IMAI, "Multilevel Coded 16-QAM Modulation with Multistage

Decoding and Unequal Error Protection", in *IEEE Global Telecommunications Conference, Globecom'98*, Nov. 1998, vol. 6, pp. 3548–3553.

[92] U. WACHSMANN, R.F.H. FISCHER, AND J.B. HUBER, "Multilevel Codes: Theoretical Concepts and Practical Design Rules", *IEEE Transaction on Information Theory*, vol. 45, no. 5, pp. 1361–1391, July 1999.

[93] K. ZHOU, J.G. PROAKIS, AND F. LING, "Decision-Feedback Equalization of Time-Dispersive Channels with Coded Modulation", *IEEE Transactions on Communications*, vol. 38, no. 1, pp. 18–24, Jan. 1990.

[94] R.E. BADRA AND B. DANESHRAD, "Asymmetric Physical Layer Design for High-Speed Wireless Digital Communications", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 10, pp. 1712–1724, Oct. 1999.

[95] A. DUEL-HALLEN AND C. HEEGARD, "Delayed Decision-Feedback Sequence Estimation", *IEEE Transactions on Communications*, vol. 37, no. 5, pp. 428–435, May 1989.

[96] M.V. EYUBOGLU AND S.U.H. QURESHI, "Reduced-State Sequence Estimation for Coded Modulation on Intersymbol Interference Channels", *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 989–995, Aug. 1989.

[97] P.R. CHEVILLAT AND E. ELEFTHERIOU, "Decoding of Trellis-Encoded Signals in the Presence of Intersymbol Interference and Noise", *IEEE Transactions on Communications*, vol. 37, no. 7, pp. 669–676, July 1989.

[98] M.J. GERTSMAN AND J.H. LODGE, "Symbol-by-Symbol MAP Demodulation of CPM and PSK Symbols on Rayleigh Flat-Fading Channels", *IEEE Transactions on Communications*, vol. 45, no. 7, pp. 788–799, July 1997.

[99] A. PICART, P. DIDIER, AND A. GLAVIEUX, "Turbo-Detection: A New Approach to Combat Channel Frequency Selectivity", in *ICC'97, IEEE International Conference on Communications, Montreal, Canada*, 1997, pp. 1498–1502.

[100] M.C. REED AND C.B. SCHLEGEL, "An Iterative Receiver for the Partial Response Channel", in *ISIT 1998, Cambridge, MA, USA*, August 16 - August 21 1998.

[101] P. STRAUCH, C. LUSCHI, M. SANDELL, AND R. YAN, "Turbo Equalization for an 8-PSK Modulation Scheme in a Mobile TDMA Communication System", in *VTC'99*, 1999, pp. 1605–1609.

[102] R. MULLER AND J. HUBER, "Iterated Soft-Decision Interference Cancellation for CDMA", in *Broadband Wireless Communications*, LUISE AND PUPOLIN, Eds., pp. 110–115. Springer, London, 1998.

[103] D.P. TAYLOR, "The Estimate Feedback Equalizer: A Suboptimum Nonlinear Receiver", *IEEE Transactions on Communications*, vol. COM-21, no. 9, pp. 979–990, Sept. 1973.

[104] W.R. KIRKLAND, *On the Application of Multi-layered Perceptrons to Nonlinear Equalization for Frequency Selective Fading Channels and*

Nonlinear Prediction for Time Selective Rayleigh Fading Channels, PhD thesis, Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, 1994.